The ISP Column

An occasional column on things Internet

May 2006

Geoff Huston

An Introduction to BGP – the Protocol

Routing in the Internet is divided into two parts – fine-grained topological detail of connected segments of the Internet is managed with interior routing protocols (such as IS-IS or OSPF), while the interconnection of these segments, or "autonomous systems" is managed by an inter-domain routing protocol, which these days is synonymous with the Border Gateway Protocol, or BGP.

In this article, the first of two articles on BGP, I'd like to cover the essential elements of BGP the protocol. In the second part next month I'd like to take a more detailed look at the current state of the Internet's inter-domain routing system.

BGP has undergone a number of refinements over its operational life. BGP was originally described in RFC 1105, in June, 1989, allowing the Internet to move on from a constrained architecture of a "core" and stub domains into a framework of peer routing domains without any central "core". BGP-2 was described in RFC 1163, in June, 1990, and BGP-3 was described in RFC 1267 in October, 1991. The current version, BGP-4, was first deployed within the Internet in 1993. The RFC describing this protocol, RFC 1771, was published in March, 1995, and subsequently refined with the publication of RFC 4271 in January 2006. The protocol has been stable for some years now and has managed the Internet routing task for over a decade. Across the deployment lifetime of BGP-4 the Internet has grown from 20,000 distinct routing entries in 1993 to some 200,000 in 2006.

An Overview of the Protocol

BGP binds together the concept of network address blocks and autonomous systems into a path vector-based routing technology. Every route object represented within a BGP-4 route database carries an address prefix and an associated path vector of AS values. BGP is essentially an inter-AS routing protocol. It does not indicate the precise path a packet should following within an AS, nor does it maintain a complete map of the topology of the Internet on a link-by-link basis. BGP uses a level of abstraction which views the Internet as a set of routing domains, or Autonomous Systems. The role of BGP is to maintain a routing map of the network at this AS level. In BGP prefixes are passed between ASes.

One of the most important route object attributes in BGP is the "AS_PATH". As address prefix reachability information traverses the Internet in the form of individual route objects, this routing information is augmented by the list of autonomous systems that have been traversed thus far, forming the AS_PATH attribute. The AS_PATH attribute allows straightforward suppression of the looping of routing information, using the simple selection algorithm that a local AS will reject any route object that already contains its own AS in the AS_PATH attribute. Also the length of the AS_PATH vector forms the BGP route path metric. A local BGP system, when attempting to select one of a number of potential route objects that refer to the same address prefix, will, in the absence of any local policy directive, prefer the route object with the shortest AS_PATH attribute length.

For example, in Figure 1 we can follow the advertisement of the prefix 10.0.0.0/8 as it is propagated within this simple inter-AS topology.

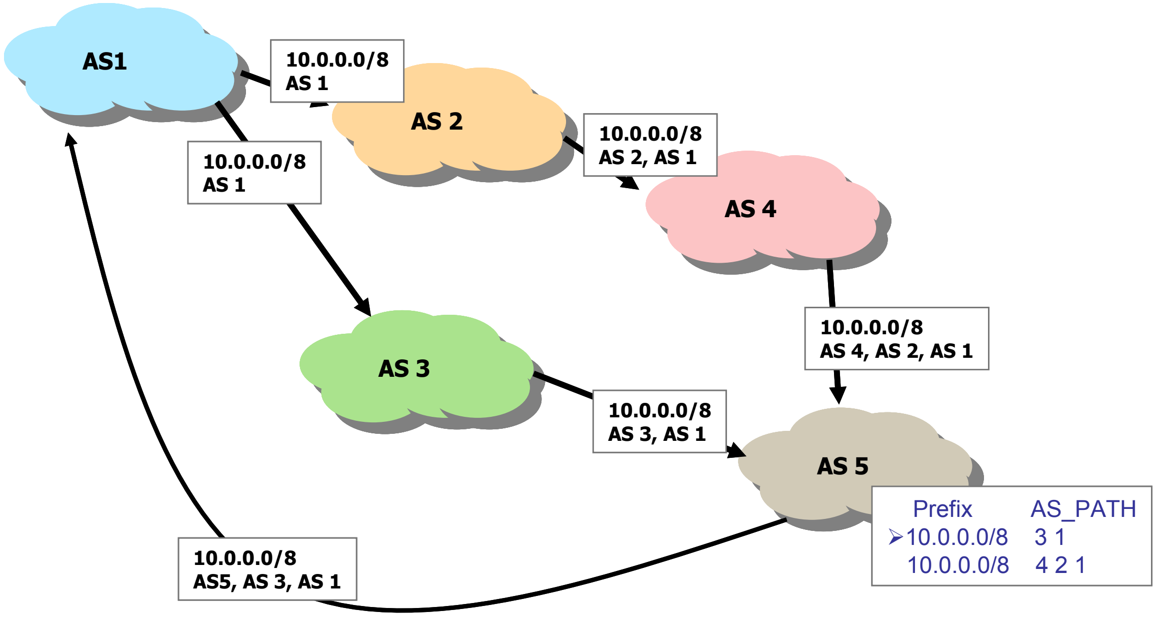


Figure 1. Multiple autonomous systems and AS paths.

In Figure 1 AS1, originating the route to network 10.0.0.0/8, advertises this prefix as route advertisement that is passed to its adjacent BGP neighbours AS2 and AS3. Both AS2 and AS3 learn the prefix advertisement with an associated path vector of <AS1>. AS2 then advertises this prefix to its neighbour, AS4. AS4 learns of the prefix with an associated path vector of <AS2, AS1>, and so on. AS5 will ultimately learn two announcements for the prefix 10.0.0.0/8, one with a path vector of <AS4, AS2, AS1> and the other with the path vector <AS3, AS1>. AS5 then makes a policy decision as to which route path to accept, which, by default will be the shorter AS path, namely <AS3, AS1>.

The general behaviour of a BGP-speaking AS is to select a single candidate path to use for a given destination address prefix, then to advertise this route to its adjacent AS peers, who learn the route with the advertiser's AS number prepended to the associated AS path vector. Loops are avoided by the simple measure of refusing to accept a route object that already includes the local AS in the object's path vector. In the example in Figure 1, AS1 can detect the AS5 announcement as a loop due to the fact that its AS value, AS1, already appears in the AS path vector.

In addition to undertaking the role of path metric and loop detector, the AS_PATH attribute serves as a powerful and versatile mechanism for policy-based routing, where a local AS can alter the default preferences for route selection based on local policy settings coupled with matching rules to be performed on the AS_PATH.

BGP-4 enhances the AS_PATH attribute to include sets of autonomous systems as well as simple lists. This extended format allows generated proxy aggregate routes to carry path information from the more specific routes used to generate the aggregate. This AS set form is used when a BGP speaker takes a number of more specific BGP routes and generates a single encompassing aggregate route. In this case the aggregator may take all the ASes used in each of the component more specific route objects and join them together in an AS set. The intent of the AS set construct is to ensure that the originators of the more specific routes, and transit ASes between the originator and the aggregation point, do not learn the aggregate route.

BGP is a member of the family of distance vector routing algorithms. While many other distance vector routing algorithms carry a simple path metric value as a scalar numeric value, BGP distinguishes itself by carrying the complete AS Path vector. The benefits of this approach is that it makes loop detection a deterministic process within BGP.

To conserve bandwidth and processing power, and in recognition of the observation that BGP is not a link-level topology maintenance protocol, BGP uses a reliable transport protocol (TCP) to support the protocol's transactions. The reliable transport implies that BGP need not explicitly confirm receipt of a protocol message, and this also obviates the need to periodically refresh the protocol state by re-flooding the entire routing information set between BGP speakers. BGP can use incremental updates, in which, after the initial exchange of routing information, a pair of BGP routers exchange only incremental changes to that information as they occur. No explicit confirmation of such protocol messages, nor any form of periodic re-flooding is supported in BGP.

BGP Messages

In the operation of BGP many of the issues about reliable information transfer within the network protocol design are addressed through the decision to use TCP as the platform for protocol communication. This design decision avoids the need to undertake explicit protocol message sequencing, acknowledgements, retransmissions and associated state timers, as TCP is able to provide BGP with a reliable transport subsystem. Of course TCP is a stream protocol rather than a record-oriented protocol, so BGP uses record marking within the TCP stream to delineate logical protocol units, or messages.

BGP uses a 16-byte marker format to delimit BGP messages. The marker is followed by a 2-byte length and a 1-byte type field, making the minimum BGP message size 19 bytes. The Marker field contains all 1's, unless a security option is being used, in which case the market contains a value based on that security option. The Length field contains the length of the entire BGP message, including the common message header, and the Type field specifies the type of BGP message. The defined Type values are:

- 1 for an OPEN message to start a BGP session,
- 2 for an UPDATE message to exchange reachability information,
- 3 for a NOTIFICATION message, which is used to convey a reason code prior to termination of the BGP session,
- 4 for KEEPALIVE messages, used to confirm the continued availability of the BGP peer,
- 5 for ROUTE-REFRESH request messages

The common header format is shown in Figure 2.

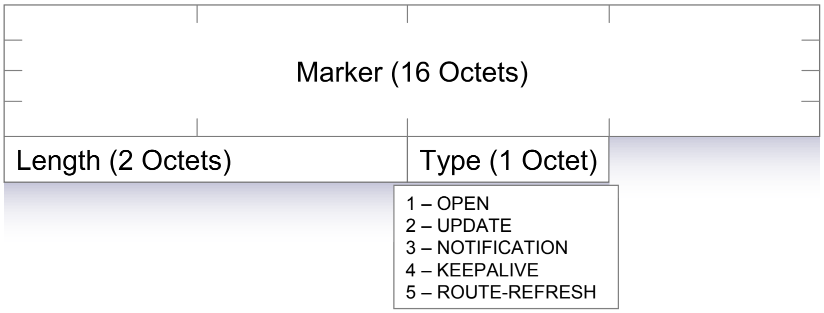
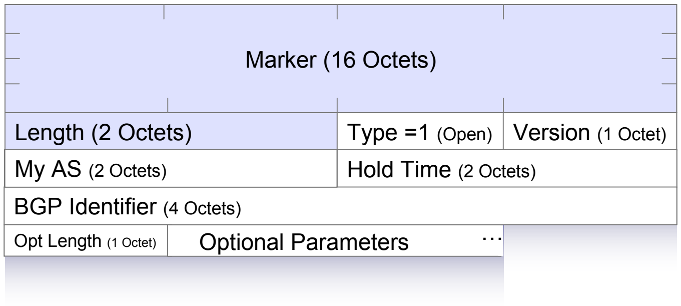


Figure 2. BGP Common Header Message Format

The protocol loads a complete routing table as part of the initial session activity and then restricts itself to sending incremental updates for the remainder of the session lifetime. There is no periodic re-flooding of the routing state, and each BGP speaker assumes that once the TCP transport session has acknowledged receipt of the message no further BGP retransmission of that instance of routing information is required. This is perhaps the most presumptive part of BGP and one of the more interesting areas of partial failure in operational deployments of BGP.

BGP uses an explicit OPEN message to commence a BGP peering session, and each BGP speaker sends its peer an OPEN message to start the BGP session. This message exchange confirms the identity of the BGP speakers by identifying the local version, identifier and local AS to each other and includes the option a capability negotiation to understand what capabilities are supported by each BGP speaker. The version field these days is usually version 4. The 'My AS" field is the 16 bit local AS of the BGP speaker, except in the case where the local AS is an unmappable 32 bit AS value, in which case the OPEN field contains the AS value 23456 and an optional capability code (65) carries the actual 32 bit local AS value. The Hold Time is the minimum 'no activity' timer. Upon its expiration the local BGP instance will assume that the BGP session has failed. The actual Hold Time for the session is the minimum of the two values. An idle BGP speaker must transmit an Update or a Keepalive message before the expiration of the Hold Timer. A zero value is interpreted as proposing no exchange of keepalive messages, which is generally discouraged as it may lead to BGP sessions that do not correctly terminate on underlying connectivity failure. The BGP identifier is a local value that is used for all BGP sessions from this BGP speaker. Typically, this is a loopback IPv4 address used by the local BGP speaker. The optional parameters negotiate authentication of the BGP session and may also contain a sequence of Type, Length, Value (TLV) triplets to negotiate session capabilities. The forms of capabilities that BGP can negotiate are listed in the IANA BGP Capabilities Registry (http://www.iana.org/assignments/capability-codes).



Marker (16 Octets)
Length (2 Octets) | Type =1 (Open) | Version (1 Octet)
My AS (2 Octets) | Hold Time (2 Octets)
BGP Identifier (4 Octets)
Opt Length (1 Octet) | Optional Parameters …

Figure 3. BGP Open Message

BGP sessions use a hold timer as the maximum amount of time between successive keepalive or update messages that can elapse before declaring the peering session inoperative and attempting to reinitialize the session. Whenever BGP sends a message, an activity timer countdown is started, using a fraction of the negotiated Hold Time as the interval timer. If this timer expires BGP sends an explicit KEEPALIVE message to ensure that the connection remains open. The hold time value is commonly set to 180 seconds. The keepalive interval is generally set to one third of the hold-time interval. Therefore, the common default keepalive transmission is every 60 seconds, so that in a stable BGP environment without updates, BGP consumes slightly less than three bits per second to maintain the connection. (Considering the constant BGP background update message noise levels of the public Internet you'd need to have the BGP speakers completely isolated from the rest of the world to get the protocol message rate down to this theoretic low level!) There is a view that this time interval is overly long, and there have been a number of proposals to reduce this time interval to one of the order of seconds or even milliseconds. The major issue here is to ensure that the BGP session is not overly sensitive to transient traffic congestion, while still reliably triggering on link level failures with a very high degree of sensitivity. The format of a KEEPALIVE message is the common header with a type code of 4 (Figure 4).

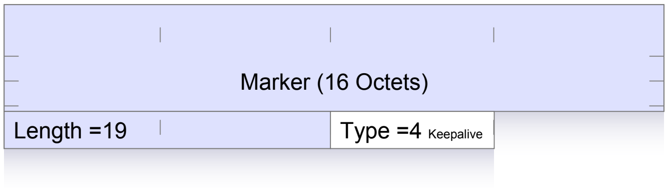


Figure 4. BGP Keepalive Message

BGP operates via the exchange of UPDATE messages. Each update message contains a set of address prefixes that are unreachable, followed by a set of common route object attributes, and then a set of address prefixes that share this set of attributes (or "Network Layer Reachability Information" (NLRI) field). (Figure 5)

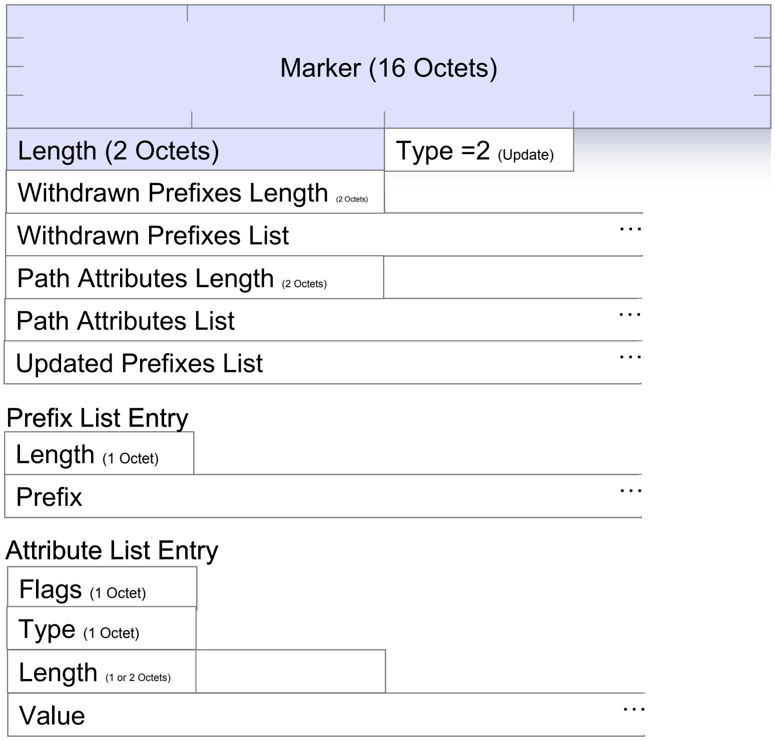


Figure 5. BGP Update Message

The unreachable, or withdrawn, prefixes are those prefixes where the local BGP instance sees no reachability to that address prefix. No path attributes are associated with these withdrawn prefixes.

The updated prefixes are those prefixes where the local BGP instance has an updated view of the reachability of a prefix, of an updated view of the attributes of the locally selected 'best' route object for a prefix. BGP is allowed to group multiple updates prefixes together in a single update message, but only if all the updated prefix share a common set of attributes.

The path attributes are a list of individual attribute flags, types and values. Each attribute may be flagged as "well-known", in which case the BGP session is closed if the attribute is not recognised by the receiver, "transitive", in which case the attribute to passed on to all other BGP peers, "partial", in which case some BGP speakers along the update path did not recognise this attribute, and "extended length' indicating that the next length field is 2 octets in length and the value is longer than 255 octets.

The commonly-used attributes used in the context of a unicast inter-domain routing system are:

- ORIGIN - Indicating whether the origin of this prefix has been learned from the interior routing protocol, an inter-AS routing session, or whether the origin of the prefix is unknown.

- AS_PATH - Describing the AS path vector associated with the prefix.

- NEXT_HOP - Describing the IP address of the border router to be used as the next hop to reach the prefix.

- MULTI_EXIT_DISC - The Multiple Exit Discriminator is used by the source AS to inform the target AS of a traffic egress point relative preference across multiple egress points between the two autonomous systems.

- LOCAL_PREF - Local preference is used to inform other BGP routers in the local AS of the originating BGP session's degree of preference for an advertised route.

- ATOMIC_AGGREGATE - Informs other BGP routers that the local system selected a less specific route without selecting a more specific route which is included in it.

- AGGREGATOR - Indicates the last AS number that formed the aggregate route and the IP address of the BGP router within the AS.

- COMMUNITY -Used to carry a set of locally defined attributes about a path.

- DESTINATION PREFERENCE - Used to bias the preference of a remote AS to a particular path.

The Notification message is used to convey the nature of an error condition prior to the closing of the underlying TCP session. The format of a Notification message is shown in Figure 6. The Error Code indicates the primary error condition, referring to an error in the message header (1), the OPEN message (2), the Update message (3), expiration of the Hold Timer (4), a state machine error (5) or a session termination (6). Subcodes provide further detail within each class of effort condition.

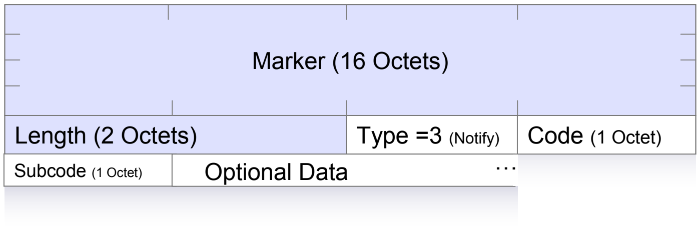


Figure 6. BGP Notification Message

A relatively recent addition to BGP was proposed in 2000 [RFC2918]. This is the Route Refresh BGP message, which request the BGP peer to re-send its peer prefixes (stored in a logical structure terms the "Adj-RIB-Out", or, in longhand the "Adjacency Routing Information Base - Outbound").This message is sent only if, during the OPEN sequence, the BGP peer has advertised a Route Refresh capability, and results in the BGP peer readvertising its Adj-RIB-Out contents. The Route Refresh message includes the address family identifier of the class of prefixes being requested in this message (AFI / SAFI codes). (Figure 7). A potential use of this command is a non-disruptive route refresh for a neighbour as an alternative to the inbound cache commonly used by BGP configurations. This is a signalling update that triggers a set of UPDATE messages that describe the contents of the matching Adj-RIB-Out.

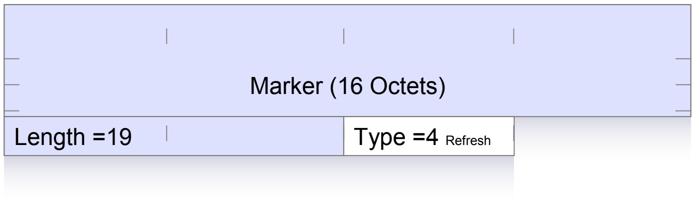


Figure 7. BGP Route Refresh Message

iBGP and eBGP

There are two forms of BGP use that differ in subtle ways. BGP is intended to provide a mechanism for one Autonomous System to exchange routes with another, and BGP sessions that connect to different Autonomous Systems are terms "eBGP" sessions. In a simple stub AS configuration, there is a single exterior boundary router that supports all the AS's eBGP sessions. The interior routing protocol typically directs a default route to this boundary point.

However, if the external connections are terminated in separate boundary routers, and the AS has a requirement to pass routes learned from one eBGP session to the other, the destination routes and associated path attributes must be passed between the two boundary routers. Using a redistribution of the BGP routes into an interior routing protocol (or IGP) to perform this transfer will cause the learned eBGP path attributes to be discarded within the IGP. An alternative approach is to set up an internal BGP peering session between the two boundary routers. Such an internal BGP session is termed an "iBGP" session. A typical application of iBGP is indicated in Figure 8.

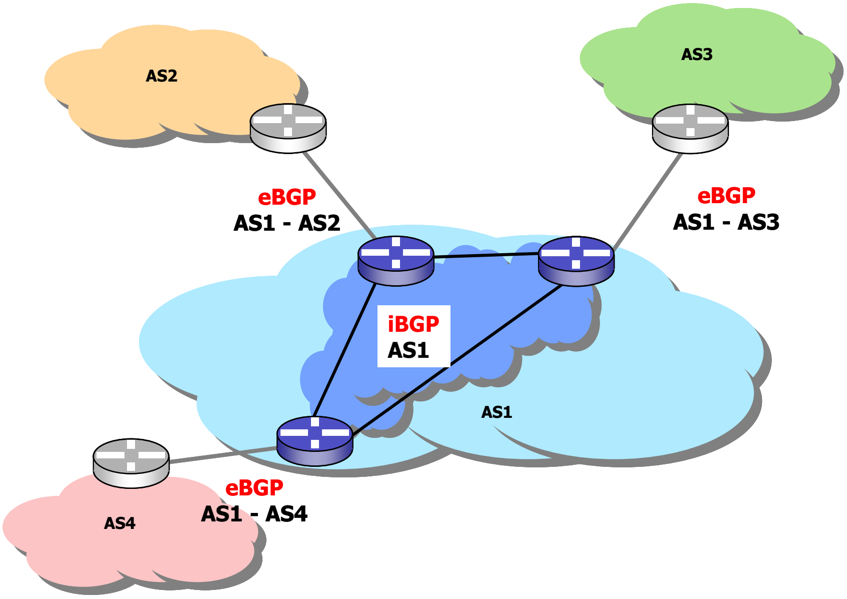


Figure 8. iBGP configuration.

Network design principles indicate that redistribution of externally learned routes into the local IGP is generally unwise. The conventional wisdom for routing design is to use iBGP as a mechanism to carry exterior-routing information along all internal transit paths between external boundary routers peers within the same AS. A default route should be injected into the interior routing system from the backbone transit-path routers, and the interior routing system can undertake the redistribution of this default route to the lower portions of the network hierarchy. In this way, packets bound for destinations not found in the interior routing table are forwarded automatically to the network's transit backbone, where a recursive route lookup reveal's the appropriate destination from the nexthop address in iBGP. At this point, internal traffic can use the exterior-routing information to make a more informed decision on how to forward traffic bound for external destinations. This deliberate omission of redistribution of BGP routes into the interior routing domain also allows for a higher degree of routing stability of the IGP.

The AS path vector construct is inadequate to detect routing loops that may arise across the iBGP sessions within the AS, so there is a simple restriction on iBGP that addresses this: routes learned via an iBGP peer session are not advertised to other iBGP peers. The corollary of this constraint is that every BGP router must form a iBGP peering session with every other BGP router within the AS. That is, all BGP speakers within a network must directly iBGP peer with all other BGP speakers within an AS.

This requirement for an N-squared peering mesh leads to one of the major scaling issues with autonomous systems and BGP. This mesh of BGP peering sessions can exceed the capabilities of the component routers, and when the iBGP mesh becomes sufficiently large, then alternative iBGP structures should be deployed. To address this iBGP load or full mesh internal peering, it is necessary to introduce some refinements to the configuration of iBGP. The most effective method is to introduce BGP Route Reflectors [RFC2796], which dilute the strict requirement for a complete mesh of peering sessions. The typical deployment structure with route reflectors is to create a small iBGP core mesh across the network's transit backbone, and configure these core iBGP routers as BGP Route Reflectors. Other iBGP peers are configured as local reflector clients from the closest core reflector. A BGP route reflector is a standard iBGP implementation, with one condition relaxed: a Route Reflector is allowed to readvertise routes learned from iBGP peers to other connected iBGP peers. A Route Reflector does not reflect all leaned routes – a Route Reflector only passes on its selected "best" path to its clients and peers. This has some implications in the time taken for a change to be propagated through a routing system and for all routers to converge to a consistent view of the network's topology.

Alternatively, the AS can be internal segmented into a number of sub-autonomous systems (typically using AS values from the private AS number space) with a BGP Confederation to create the external appearance of a single AS. This segmentation may have implications for any non-transitive path attributes that may be deployed, and it is less disruptive to use iBGP Route Reflectors as the mechanism to address iBGP scale issues. These configurations are indicated in Figure 9.

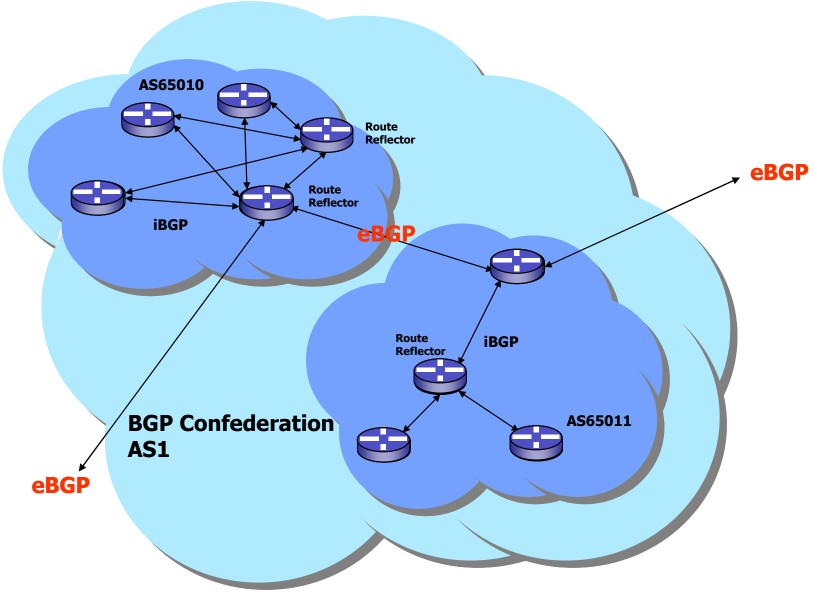


Figure 9. iBGP route reflectors and BGP confederations.

BGP Route Selection Process and Routing Policies

The default BGP route selection process is to prefer a path with the most specific network address that matches the destination address. This is a generic classless matching algorithm, in which the longest prefix is the preferred match. When comparing two route objects that refer to the same prefix, then there are a sequence of comparisons to determine which route object is selected by the local BGP speaker

1. Select the route object with the highest value for LOCAL-PREF
2. Select the route object shortest AS_PATH
3. Select the lowest MULTI_EXIT_DISCRIMINATOR
4. Select the minimum IGP cost to the NEXT_HOP address
5. Select eBGP over iBGP-learned routes
6. If iBGP select the lowest BGP Identifier value.

However, a number of approaches can be used to alter this path selection process.

Some of these approaches are local mechanisms that can be undertaken only by the network administrator, affecting the path taken for traffic leaving the AS. Typically these are configuration settings that alter the LOCAL_PREF value based on locally set criteria.

Other mechanisms are passed to a remote AS and attempt to bias the remote AS's egress path selection policy to match with the preferred local AS's path ingress policy. The most generic method to bias the path selection in a remote AS is to alter the AS_PATH length, by artificially increasing the length of the least preferred paths for incoming traffic.

Such an approach may not necessarily have a definitive outcome, because a remote AS can also apply a preference as to which egress path should be selected. The local AS may choose to implement local path selection policies, which may override these remote indications. Because the relative complexity of the transit structures used within the Internet today, such efforts of exerting control at a distance involve not only the two autonomous systems attempting to negotiate an agreed policy, but also involve the consideration of the path selection policies of all autonomous systems positioned on a potential transit path between these two autonomous systems. Consequently, inter-AS routing has often been described as an art rather than a science! Of course the one relatively universal rule is that the ore specific path is preferred over that of a covering aggregate, and the prevalence of more specifics on the inter-domain routing table bears witness to the perceived efficacy of this approach, despite the cost in a higher routing overhead for all.

AS Path Filtering

One of the most obvious ways to bias the BGP path selection process is to omit from consideration those route objects that do not conform to the policy of the local AS via some filter mechanism. If a route is not seen for a particular destination from a particular peer, traffic will not be forwarded along that path.

For example, in the network of Figure 10, AS3 normally would pick a path to AS2 of the form (AS1, AS2), preferring it to the longer path of (AS5, AS4, AS2). However, AS3 may have negotiated a better transit price from AS5 and does not want to use AS1 as a transit. Instead, AS3 wants to present all traffic destined to AS2 through AS5 as the first hop AS. In this case, AS3 could filter all incoming paths of the form (AS1, AS2) from AS1's routing advertisements to AS2, to allow AS2 to conform to its routing policy.

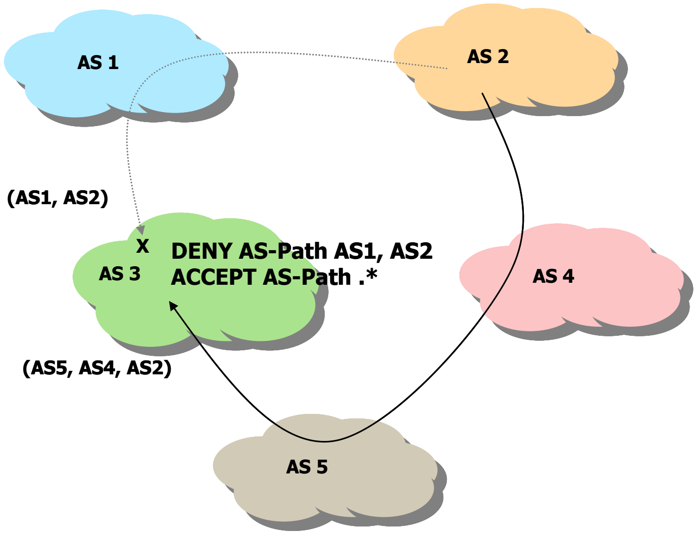


Figure 10. AS path filtering.

The filtering can be done on an inbound (accept) or outbound (advertise) basis, based on any variation of the data found in the AS path attribute. Filtering can be done on the origin AS, any single AS, or any set of autonomous systems found in the AS path information. This provides a great deal of granularity for selectively accepting and propagating routing information in BGP. By the same token, however, if an organization begins to get exotic with its routing policies, the level of complexity increases dramatically, especially when multiple entry and exit points exist. The downside to filtering routes based on information found in the AS path attribute is that it really provides only binary selection criteria; you either accept and propagate the route, or you deny and do not propagate the route. This does not provide a mechanism to define such subtleties as conditional filters, such as are used to create a primary and backup path.

AS Path Prepending

In referring back to the network in Figure 10, inbound routing filters have had the result of causing AS3 to direct its outgoing traffic to AS1 via AS5. However, traffic flow in the opposite direction may not obey the same policy. AS2 will see for AS3 a path of (AS1, AS3) compared to an alternative path of (AS4, AS5, AS3) and will, by default, select AS1 as transit to reach AS3. Can AS3 affect the path selection properties of AS2 to bias it to select the longer path?

The AS paths can be manipulated to achieve this outcome using a mechanism called AS path prepending, which is the practice of inserting additional instances of the originating AS into the beginning of the AS path prior to announcing the route to an exterior neighbour. If AS3 inserted two additional instances of AS3 into the AS path advertised to AS1, then AS2 would be faced with selecting a path between (AS1, AS3, AS3, AS3) and (AS4, AS5, AS3), and the desired result would be achieved. This outcome is indicated in Figure 11.

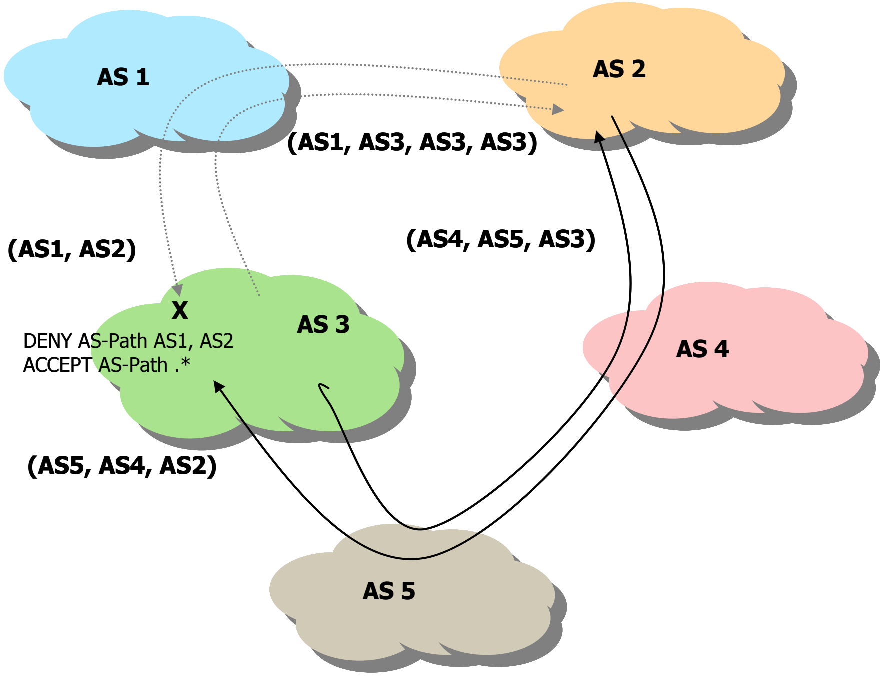


Figure 11. AS path prepending.

When used in isolated instances, AS Path prepending can produce the desired result, but when used in a more widespread fashion AS Path prepending can cause constant adjustment rippling across the network. In this example network, if AS4, in order to satisfy a local policy requirement of its own, decides to perform AS path prepending on all routes announced to AS2, then AS2 may be presented with a choice between <AS1, AS3, AS3, AS3> and <AS4, AS4, AS4, AS5, AS3>, and the path selection will flip back to transit AS1, without AS3's knowledge.

Similar to AS path filtering, AS path prepending is a negative biasing of the BGP path-selection process. The lengthening of the AS path is an attempt to make the path less desirable than would otherwise be the case. This mechanism commonly is used for defining candidate backup paths. AS path prepending cannot positively bias the path selection process.

Specific Route Injection

BGP always will prefer more specific routes, irrespective of the relative AS path lengths of the specific and more general route objects. This preference can be used to bias path selection and to undertake primary and backup paths for particular routers without having to undertake AS path prepending.

In the example network indicated in Figure 12, if AS3 wanted AS2 to use the path (AS4, AS5, AS3) for the network address 10.0.0.0/8 and use (AS1, AS3) as a backup path for this network address, then AS3 could advertise 10.0.0.0/8 to AS2 and advertise the routes 10.0.0.0/9 and 10.128.0.0/9 to AS5. AS2 will use the longer path as its preferred path due to the existence of more specific routes within its routing table. The AS path length is examined only if identical prefixes are advertised from multiple external peers.

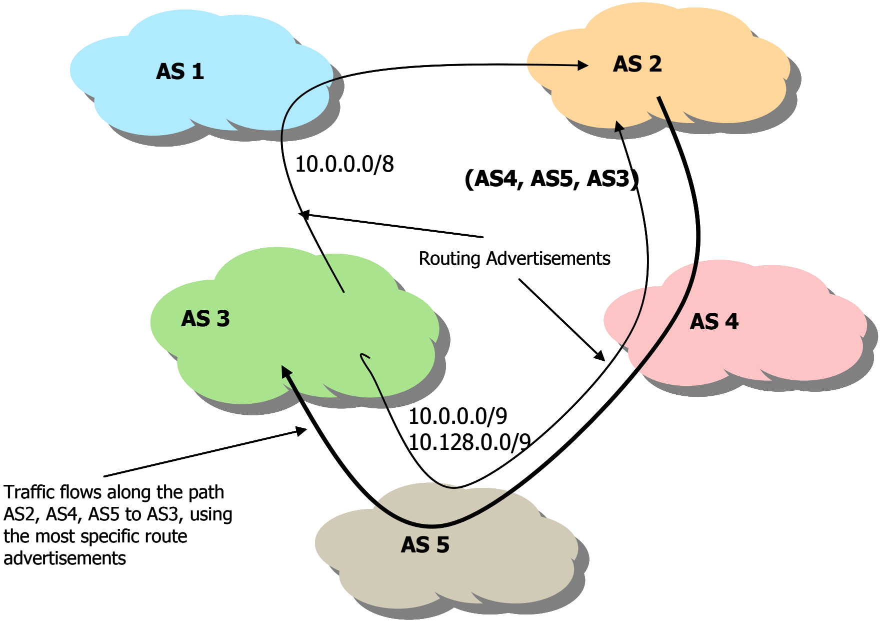

AS 1
AS 2
10.0.0.0/8
(AS4, AS5, AS3)
Routing Advertisements
AS 3
AS 4
10.0.0.0/9
10.128.0.0/9
Traffic flows along the path AS2, AS4, AS5 to AS3, using the most specific route advertisements
AS 5

Figure 12. Specific route selection.

Specific route injection should be used with extreme caution. One of the major scaling issues the Internet faces as a whole is the continued growth in size of the Internet routing table. Widespread use of specific route injection as a means of biasing routing policy would cause new pressures on the size of the devices used to carry Internet routes. Already, some backbone transit providers enforce entry filters that block specific routes from well-defined ranges of network addresses, in an attempt to limit the amount of fragmentation in the routing space. Overuse of specific route injection would cause further measures to be adopted to ensure that the global routing table remains within a workable size.

BGP Communities

Another method of making routing policy decisions using BGP is to use the BGP community attribute to group destinations into communities and apply policy decisions based on this attribute instead of directly on the prefixes. A number of defined communities have a determined outcome, and others can be defined by network operators.

The most well-known (and widely used) community is that of No_Export. When a path attribute includes this community, the route cannot be advertised to peers outside of the local confederation. In the network of Figure 13, another way to ensure that AS2 does not transit AS1 to reach AS3 is for AS2 to mark the BGP advertisements to AS1 with the No_Export community attribute. This attribute prevents AS1 from advertising the routes to AS2.

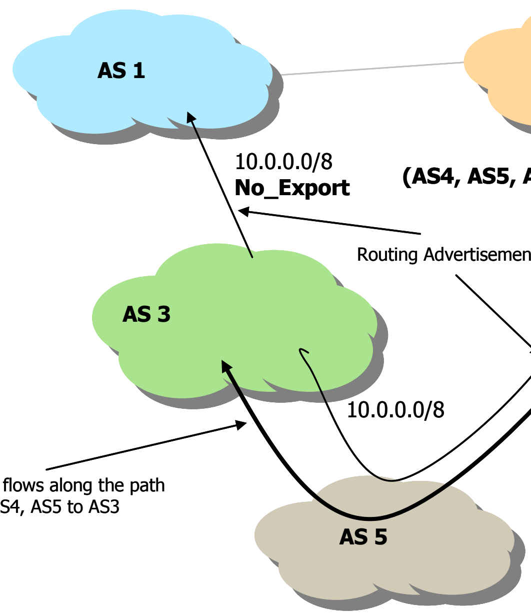


Figure 13. BGP communities.

Communities provide a useful and convenient mechanism for an AS to place tags on routes. Consequently, this mechanism has grown in popularity because it provides a simple and straightforward method with which to apply policy decisions relating to the treatment of routes.

As indicated in Figure 14, AS1 has a customer, AS2, and participates in two peering exchanges, in which it peers with AS3 and AS4. The policy requires that AS1 does not want to act as a transit for AS3 to reach AS4, but does act in a transit capacity for its client, AS2. AS1 could place all routes received from AS2 into a community 1:10, all routes received at the peering exchanges into a community 1:20, and transit routes into the community 1:30. At the exchange, AS1 announces only routes that include the community attribute 1:10; thereby not announcing routes that would make AS1 act as an unfunded inter-exchange transit operator.

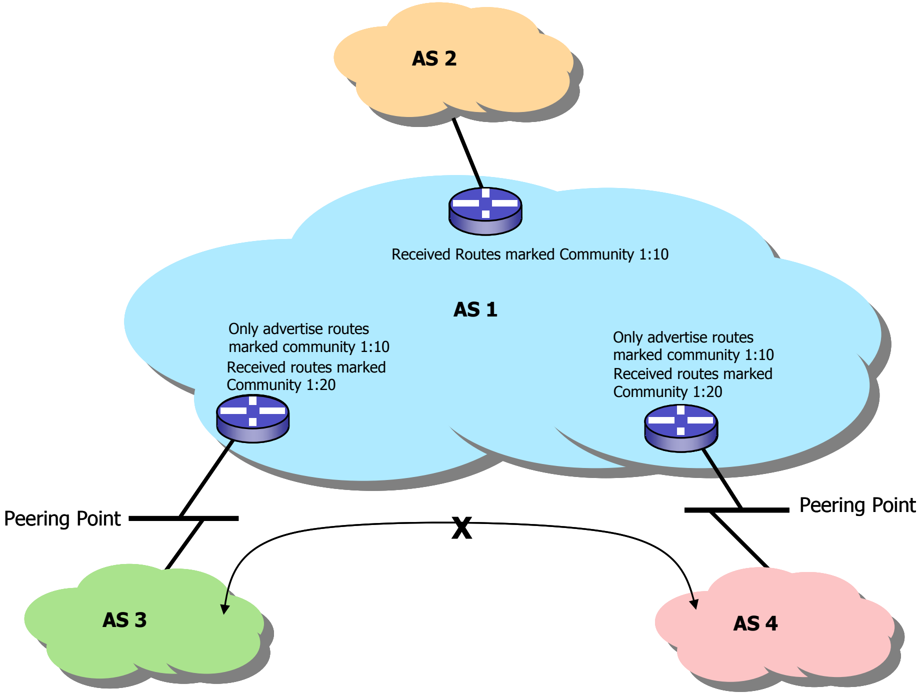


Figure 14. BGP communities and peering.

BGP communities are very flexible tools, allowing groups of routing advertisements to be grouped by a common attribute value specific to a given AS, as the community value has two parts: a two-byte AS number and a two-byte value. The community can be set locally within the AS or remotely to trigger specific actions within the target AS. The community value can be defined by a provider to allow remote autonomous systems to label advertised routes as primary or backup or to allow the default path selection process to be weighted by a community attribute. Community values can trigger actions that encompass more than path selection and manipulation and can be defined to modify any of the actions of the BGP routers.

BGP Local Preference

BGP Local Preference is a local path attribute (local to the immediate AS), which indicates the preference given to the route object by the AS. BGP local preference is carried only within iBGP sessions, so it applies across the entire AS, but the attribute is non-transitive, or rather, not passed along to neighbouring autonomous systems.

An example of the use of the local preference attribute is indicated in Figure 15. Both AS2 and AS3 are advertising a route to 10.0.0.0/8, as AS4 is multiply homed. AS1 can make a decision to prefer the path via AS2 by setting the local preference attribute to 20 at boundary router A for the 10.0.0.0/8 route being presented by AS2 and by setting the local preference attribute to 10 at boundary routers B for the incoming 10.0.0.0/8 route. The iBGP sessions ensure that these local preference settings are promulgated throughout the BGP routers within the AS, ensuring that they will select the path directed towards AS2.

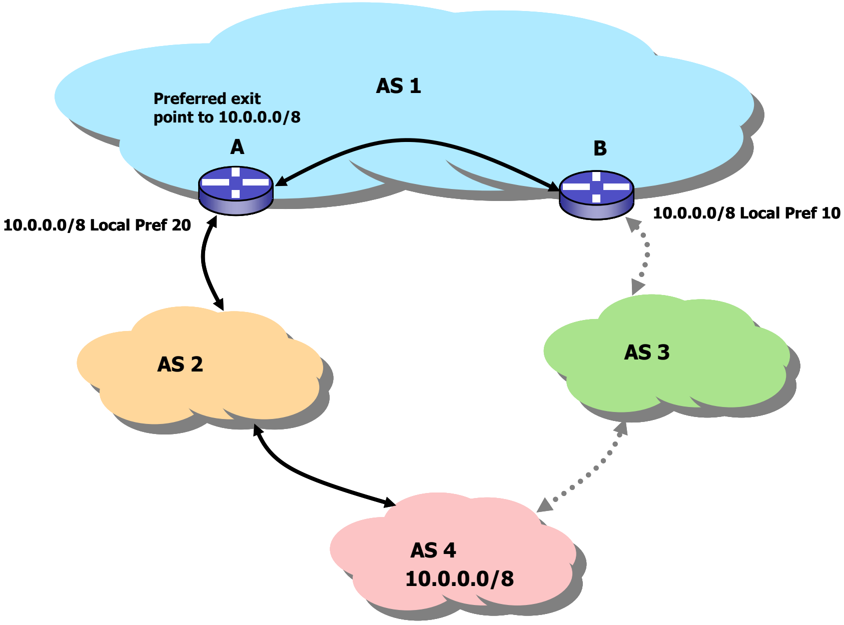


Figure 15. BGP local preference.

The Multi-Exit Discriminator (MED) Attribute

Another BGP tool that can be used to bias route selection policies between autonomous systems is the Multi-Exit Discriminator (MED) path attribute. The difference between a MED and a local preference attribute is that while the local preference attribute is a local tool to select an outgoing path, the MED is an exported attribute to inform adjacent autonomous systems of a preferred ingress path to the AS. The MED attribute is passed to the neighbouring AS and no further.

The primary use of the MED path attribute is to allow the network administrator to inform an adjacent AS of a preferred ingress path for the route when multiple links exist between the two autonomous systems, with each link advertising the same length prefix. MED is a route-specific tie-breaker in such cases and can be used on a route-by-route basis to allow the network administrator to load balance incoming traffic across multiple links. In Figure 16, AS1 has two connections to AS2 and wants different links used for particular route objects. For incoming traffic addressed to network 10.0.1.0/24, AS1 prefers that path A is used by AS2, and for network 10.0.2.0/24, AS1 prefers that path B is used by AS2. Check to ensure that the adjacent AS honours the advertised MED, as the AS may use a local export policy that may override the MED attribute setting, before extensive setting of MED values to implement load balancing.

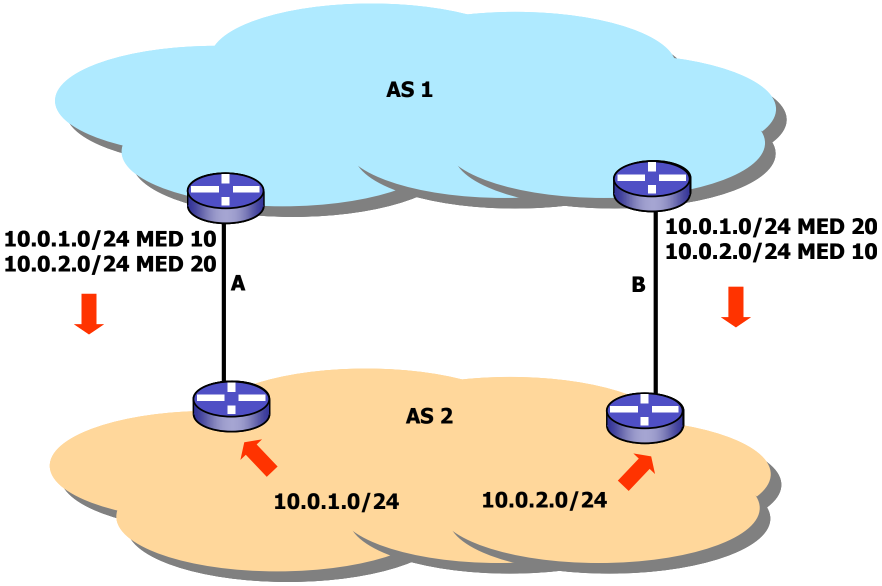


Figure 16. BGP Multi-Exit Discriminator.

BGP Route Damping

One of the major performance issues within the BGP environment is the level of route change. Within the Internet, every route change in the network that causes a BGP-announced address prefix to be withdrawn, or re-announced with different attributes, causes a BGP update to be propagated across the network. Given the properties of distance vector algorithms it is also common to see a single BGP event, typically a withdrawal, generate an entire set of update and withdrawal BGP messages spanning up to minutes. The resulting router load in computing consequent changes to the routing table within the Internet can be overwhelming, particularly when a route entry starts to “flap”. A flap is a rapid oscillation of route withdrawal followed by route re-announcement. It can be caused by route instabilities elsewhere in the network or by faulty circuits that have an error rate which is just marginal to support the operation of BGP. Route selection, and indeed route-entry promulgation should take into account a route entry's history of flapping and attempt to avoid the use of routes that are exhibiting instability.

Some implementations of BGP allow the network operator to configure BGP to remove route entries from consideration while the entry is flapping and will reconsider the route entry for inclusion in the route selection process only after it has remained stable and available for a sufficiently long period. Damping implementations assign a route a penalty value upon each flap, and when the penalty exceeds a threshold value, the route is suppressed, no longer considered in path selection, and no longer propagated to neighbouring autonomous systems. In the absence of further flaps, the penalty value exponentially decays. Further flaps increase the penalty value. When the penalty value falls below a reuse threshold, or the route has been suppressed for a sufficiently long interval, the route is reconsidered.

The problem here is that BGP itself becomes a “flap” generator when a route is withdrawn. As the network interconnects more densely then the level of BGP “noise” of a sequence of observed withdrawals and updates arising from a single original withdrawal gets larger. This has prompted some reconsideration of the wisdom of using flap damping, and the current thinking appears to be that, as currently implemented, BGP route flap damping is to be avoided as, to quote a recent RIPE document on this subject, “If flap damping is implemented, the ISP operating that network will cause side-effects to their customers and the Internet users of their customers' content and services as described in the previous sections. These side-effects would quite likely be worse than the impact caused by simply not running flap damping at all.” [RIPE Routing Working Group’s Recommendation on Route-Flap Damping”, RIPE-378, May 2006]

AS Paths and AS Sets

AS paths are normally an ordered sequence of AS values, which are intended to describe the sequence of autonomous systems that a packet must transit to reach the destination using this path. Conventionally, an AS Path also describes the sequence of ASes that the BGP update message has traversed in order to reach the local speaker. The AS Path is used in three ways in BGP. The first is as a path length metric, where the path metric associated with a prefix is the number of entries in an AS Path (an AS Set counts as 1 for the purposes of a path metric). The metric does not care about duplicated AS numbers in the AS Path, so a common operational technique to make an AS Path appear longer than it would otherwise appear is to ‘prepend” a number of instances of the local AS Number to the AS Path. The second use of the AS Path is for loop prevention. If an AS detects its own value in the AS Path it discards the update as a routing loop. This avoids the conventional ‘count to infinity problem of loop detection in distance vector protocols that do not maintain an explicit path vector. Lastly the AS Path can be used as a policy input, where a local AS can express various local path selection preferences based on the AS Path value.

BGP allows the use of prefix proxy aggregation to combine a number of specific routing entries into a more general aggregate single routing entry that encompasses the specific routes. Within this operation, some AS path detail may be lost, which, in turn, could admit the possibility of routing loops forming. At the point of this route aggregation, the local AS forms an initial AS path for the new route, which contains a single element, an AS set. This set is constructed by forming the union of the AS paths used in all the component route entries, and the resulting set is an unordered collection of AS values.

Figure 17 indicates AS4 aggregating the route 10.1.0.0/17 originating within AS1 with the path (AS2, AS1) with a second route, 10.1.128.0/17 originating within AS3 with the path (AS3). The resulting AS path of the aggregate route, 10.1.0.0/16 ,is the union of these two paths, {AS1, AS2, AS3}, and the path advertised to AS5 is (AS4, {AS1, AS2, AS3})

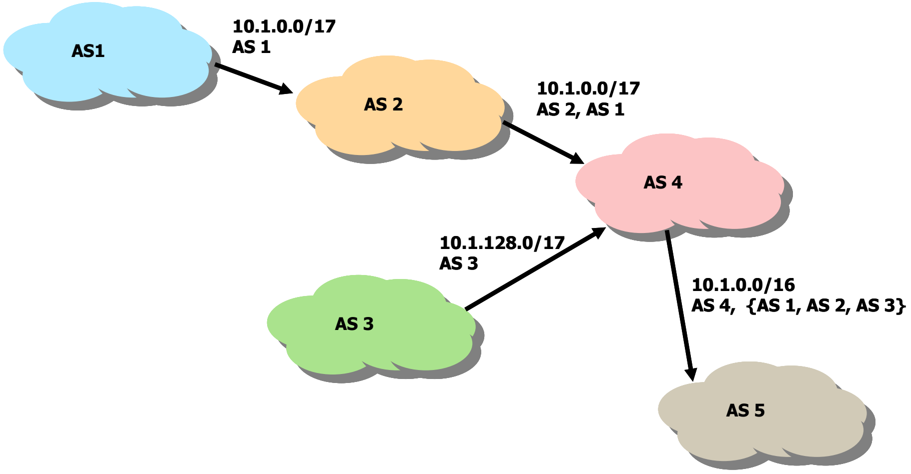


Figure 17 Aggregating Routers and AS-sets.

eBGP Multihop

The most straightforward method of connecting two boundary BGP routers is via a direct physical connection between the two devices. This connection can be a point-to-point circuit, or it can take the form of a peering across a local broadcast network. The latter is the common case for Internet exchanges and similar peering constructs.

This connection is not always possible to achieve, and configurations in which the external connection is load-shared over a number of parallel circuits or in which intervening routers are not boundary BGP routers. In such situations BGP allows the use of a multihop IP connection between the two boundary routers.

BGP Session Security

It should be noted that holding up a very long lived TCP session across a network path presents a number of unique security risks, and BGP is certainly vulnerable in this respect. One of the more disruptive forms of attack of such BGP sessions is attempting to guess the current TCP sequence number window and injecting a gratuitous TCP Reset packet into the session. If the TCP sequence number of the injected reset packet is within the valid TCP window this will cause the BGP session to close, the routes to be withdrawn, and then the routes to be re-propagated upon session restart. If this can be undertaken repeatedly this can form a very effective denial of service attack. There are a number of ways to mitigate this risk. One approach is the BGP TTL Security Hack, which specifies a minimum value for the TTL of received BGP packets. This implies that this form of hostile injection must be performed within the same IP hop count radius as the two BGP speakers (this approach is written up in a now-expired internet-draft: draft-gill-btsh).

An approach which is more resistant to injection attacks is the MD5 Signature Option [RFC2385], which adds a 16-byte MD5 digest to every TCP packet in the BGP session as a TCP option. The MD5 digest encompasses the TCP pseudo header, the TCP header, the TCP payload and a connection-specific key value. Using this makes the BGP session relatively resistant to most forms of packet injection, although it has been noted that this still does not provide per-packet authentication, integrity, confidentiality or replay protection. This implies that there are still some issues there that may benefit from a more robust solution.

Such an approach is to use IPSEC for session protection (this is also documented in a now-expired internet-draft, draft-ward-bgp-ipsec). While this provides for higher levels of assurance that the BGP session is resistant to packet injection attacks, there are some concerns over the incremental workload being placed on the routers in terms of the cryptographic workload and the time criticality of BGP convergence.

The other pragmatic observation about BGP security is that it appears that by far the most straight form of attack is to obtain control and configuration access to a deployed router and use this compromised platform as the base for launching attacks on the routing system. In the face of such an encompassing attack on the control instruments of the routing system, BGP session-level security needs to be placed in the appropriate perspective. (See the ISP Column articles “Securing Routing – An ISP View” (February 2005), and “Securing Inter-Domain Routing” (March 2005) for a more detailed look at this topic.)

And in the Next Column

So how well does BGP actually work? Does this Distance Vector Protocol actually operate efficiently in something as large and diverse as the Internet? And how well can BGP scale to meet tomorrow’s demands? By building on this description of the workings and application of BGP, I’ll try to provide some answers to these questions next month.

Disclaimer

The views expressed are the author's and not those of APNIC, unless APNIC is specifically identified as the author of the communication. APNIC will not be legally responsible in contract, tort or otherwise for any statement made in this publication.

About the Author

Geoff Huston B.Sc., M.Sc., has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of a number of Internet-related books, and has been active in the Internet Engineering Task Force for many years.

www.potaroo.net