BGP

The Autonomous System (AS)

Although with routing protocols such as OSPF and EIGRP we talk of ASs, these ASs mean purely routing domains that use different IGPs. When we talk of ASs in the more global sense, then we are talking about ASs that are under different administrations, where we do not have the control on policies that we would if we were managing a group of internal ASs.

IDLP is BGP as implemented by ISO. The **Internet Assigned Numbers Authority (IANA)** now gives out Autonomous System (AS) numbers which range from 1 to 65,535. Any AS numbers between 64512 and 65535 are for private use. RFC 1930 gives guidelines as to the use of AS numbers.

Nowadays, ASs may be **Single-homed (Stub)**, **Multi-Homed** or **Transit** ASs. When you need to pass traffic through your network from one or more ASs or you need to manage the traffic coming in from a particular AS, BGP is often necessary. ISPs typically use BGP. BGP is good at prevent looping, along with better advertisement of tens of thousands of routes and for better administration of routing policies.

Really, a Single-homed AS need only have a default route advertised internally, and the ISP need only advertise the internal AS network if it is not already part of the ISP's address space. No routing protocol would be required.

You can have a Multi-homed AS where one link to a particular ISP acts as backup to a higher bandwidth link to the same ISP. Again using default routes where the backup route has a higher administrative distance would be fine here.

Having a backup link is not very efficient, a better solution is to load share across two links and enable each link to back up the other. Using OSPF is a good way to do this since both default routes can be advertised into the AS with equal costs and as External Type 1s. Routers within the AS take into account the internal cost of a route that is an External Type 1 (E1) to the ASBRs. This results in internal routers using the nearest exit points, thereby loadbalancing traffic.

These solutions so far do not require BGP, but if you wanted to have more control on routes that are to be advertised and to modify metrics associated with these routes, then BGP may be more suitable. An example is when you are multihoming to multiple ASs where you have to advertise routes through different ISPs that own different blocks of addresses. These ISPs are unlikely to want to coordinate with each other, let alone 'punch holes' in their address blocks or advertise small address spaces (see a little later).

Ultimately, load-balancing when connecting to ISPs is not precise because you cannot control the quality of access of the ISPs and beyond. Instead, multihoming should primarily be used for resilience.

IGPs and EGPs

IGPs (Interior Gateway protocols) use metric interface costs (OSPF) or hop counts (RIP) to determine the best paths. **Exterior Gateway Protocols (EGPs)** link varying IGPs and use administered routing policies to determine best paths through service providers.

Originally **EGP** was used with the old Internet topology which, due to its small size, was a simple two tier model with a core AS and the additional ASs around it. An AS was given a 16-bit number and every 3 minutes EGP advertised the routes that it knew with other EGP peers via a full class IP address (no subnets) and a metric from 1 to 255, with 255 being unreachable. EGP is considered obsolete except in large private networks.

The main problem with EGP is that it could not cope with a meshed network of ASs, EGP could not detect loops and had no way of creating policies for routing. EGP was merely a reachability protocol rather than a routing protocol.

The Internet has grown substantially and now has a very hierarchical structure which can be summarised thus:

- **Subscribers**
- **Local ISPs** - Tier III
- **Regional Service Providers** - Tier II
- **Network Service Providers** - Tier I
- **Network Access Points (NAP)** - these interconnect the Tier I providers.

NAPs use Unix route servers running BGP. They share the **Routing Arbiter Database** of BGP routes which is copied between route servers.

Border Gateway Protocol (BGP) was the replacement for EGP and is not strictly a routing protocol, it is often described as a distance vector protocol because it uses path vectors, it is more policy-based than RIP. BGP knows nothing of what goes on within an Autonomous System (AS) it is used to *link* ASs, guaranteeing a loop-free environment.

BGP-1 was first defined in RFC 1105 in 1989, then was updated with BGP-2 in 1990 with RFC 1163 only to updated again to BGP-3 in 1991 with RFC 1267.

The current version of BGP is known as BGP-4 and was defined in RFC 1654, RFC 1655, RFC 1771 and RFC 1772. BGP-4 was different from earlier versions in one main respect and that was it became a classless routing protocol and thereby supported CIDR. For information on CIDR have a look at CIDR.

CIDR does have a number of limitations:

- Lack of portability - an ISP will give you a CIDR block that is part of a larger CIDR block owned by that ISP. If you want to change ISP then you are unlikely to be able to keep the CIDR block. Having to change the IP addresses of the end user devices is made easier by using DHCP and/or NAT with private addressing. If you are an ISP and you want to change your upstream provider e.g. a Regional ISP, then you not only have to deal with your own addressing problems but also those end users that depend on you for access.
- Lack of flexibility when connecting to multiple providers - in order for Internet traffic to reach your small CIDR from both ISPs, you must advertise that block to both ISPs. If the CIDR block is say /22, then the rest of the world can see this block through two different ISPs and will use the more specific route to this network. The original ISP that 'owns' this block has it included as part of a larger block and only this larger block is advertised out. It is possible for the more specific /22 route to be re-advertised back into the originating ISP. To stop this the ISP that owns the block must advertise the more specific /22 block as well as the larger block i.e. punch a hole in the block. This is not desireable. Even if the ISP does agree to advertise the /22 block, most Tier 1 providers do not accept any blocks less than /19 (called a **Globally Routable Address**) in order to minimise the routes in their backbone. To help with the address dependency problem, it is possible to obtain **Provider Independent Address Space**

which is portable and is not dependent on the ISP. This does not help however if the address space is smaller than /19 and is therefore not accepted by Tier 1 providers.

BGP-4 Overview

BGP is considered to be a 'Path Vector' routing protocol rather than a distance vector routing protocol since it utilises a list of AS numbers to describe the path that a packet should take. This list is called the **AS_PATH**. Loops are prevented because if a BGP speaking router sees it's own AS in the AS_PATH of a route it rejects the route.

A router in a transit AS may have extremely large routing tables (up to 90,000 networks amounting to over 30Mb) and BGP-4 uses **Classless InterDomain Routing (CIDR)** to slow the growth of these tables.

The router maintains routing tables for the IGP as well as the BGP and information can be exchanged between them.

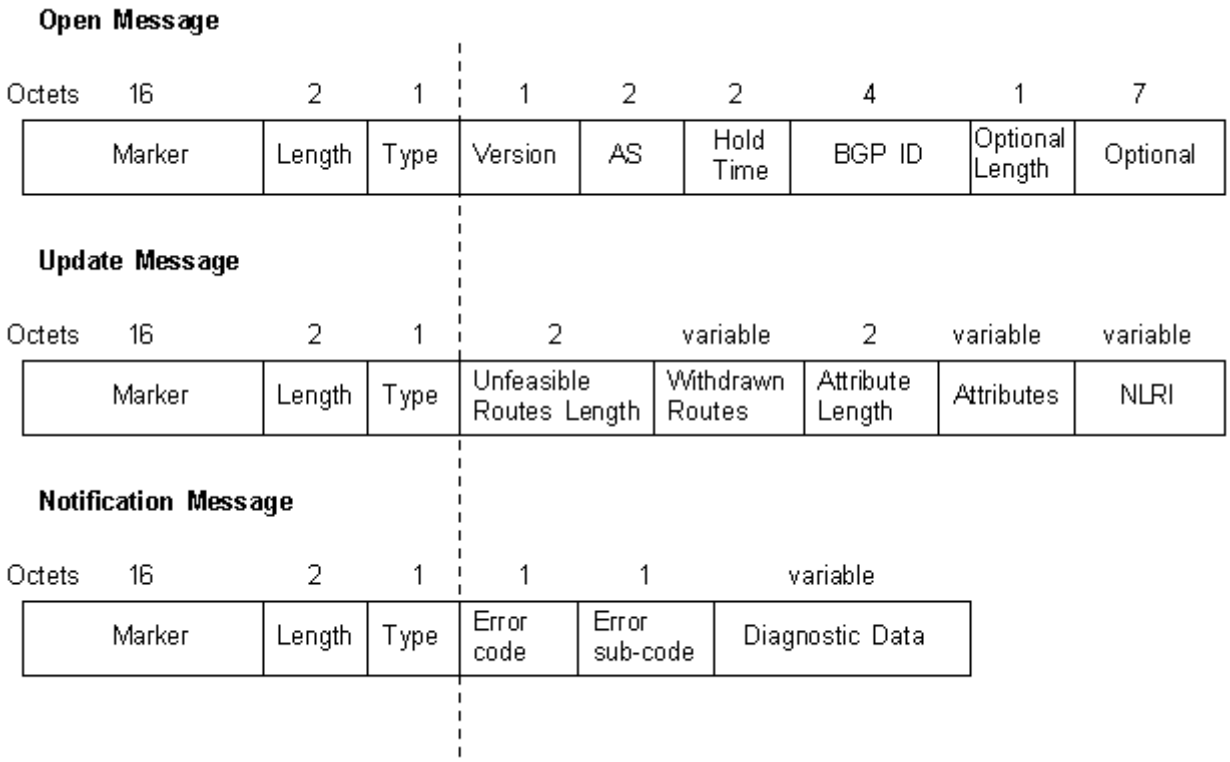
There are two types of sessions between a router and its neighbours:

- **External BGP (EBGP)** sessions occur between routers in different ASs which are usually next to each other sharing the same media and subnet.
- **Internal BGP (IBGP)** sessions occur between routers within the same AS and these sessions are used to synchronise the routing policy within an AS. These routers do not have to be next to each other however they do need to be able to see each other so that a TCP connection can be made between them! You would configure these if you were needing to pass BGP information to other ASs.

BGP-4 uses TCP (port 179) for sending and receiving messages reliably between **Peer routers**. (BGP calls routers **Speakers**, and routers that run between each other are called **Peers**). The reliable connection means that only changes need to be sent between peers rather than complete tables. These updates can be triggered updates rather than periodic updates. Only keep alive messages are sent regularly.

BGP Message Structure

The BGP message varies between 19 and 4096 octets in size and has the following structure:



On this newly formed TCP connection the following list describes the types of messages with the code numbers:

1. **Open message** (code 1) - containing the BGP version number, the originating AS, the Hold time and the BGP router ID. This ID is by default the highest IP address on the router (as in OSPF).
2. **Keepalive** (code 4) - this follows the acceptance of the open message as it is sent back and consists of just the 19 octet message header. The keepalives are sent every 60 seconds just to stop the hold down time from expiring.
3. **Update message** (code 2) - one of these is required for each path. This contains one or more **Network Layer Reachability Information (NLRI)** listed as IP (Length, Prefixes) tuples, any withdrawn IP (Length, Prefixes) tuples (CIDR supernets) and Path attributes associated with the NLRI.
4. **Notification Message** (code 3) - this is sent if there is an error and always closes the connection.

The following list details the possible Error codes and Error Subcodes in the Notification Message:

1. **Message Header Error** for which the possible Error Subcodes are:
    1. Connection not synchronised
    2. Bad message length
    3. Bad message type
2. **Open Message Error** for which the possible Error Subcodes are:
    1. Version number not supported
    2. Bad peer AS
    3. Bad BGP ID
    4. Optional parameter not supported
    5. Authentication failed

6. Hold time not accepted

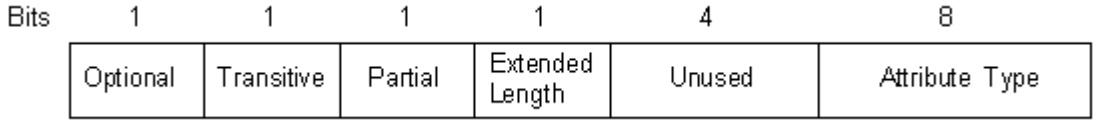
3. **Update Message Error** for which the possible Error Subcodes are:
    1. Attribute list corrupted
    2. Well-known attribute unrecognised
    3. Well-known attribute missing
    4. Attribute flag error
    5. Attribute length error
    6. ORIGIN attribute incorrect
    7. AS routing loop
    8. NEXT_HOP attribute incorrect
    9. Optional attribute error
    10. Network field incorrect
    11. AS_PATH incorrect
4. **Hold Timer expired**
5. **Finite State Machine error**
6. **Cease**

The BGP connection has 6 possible states:

- **Idle** - the router waits for a Start Event from either a new BGP process being configured or being reset before it initialises a TCP connection and starts the ConnectRetry timer which defaults to 60 seconds.
- **Connect** - the router waits for the completion of the TCP connection. Once complete, the router resets the ConnectRetry timer and sends an Open message to the neighbour.
- **Active** - this the state of the router that is initiating a TCP connection i.e. sends a Start Event. Again the ConnectRetry timer is used.
- **OpenSent** - The Open message has been sent and the router is waiting for an Open Message from its neighbour. The Keepalive is sent next and the Hold Time is negotiated down to whichever router has the lowest value.
- **OpenConfirm** - the router is waiting for a Keepalive or Notification message.
- **Established** - once a Keepalive or Update message is received the Hold time is started and the BGP peer connection has started.

Path Attributes

In the Update message, Path attributes are sent in triplets which consist of **Attribute Type**, **Attribute Length** and **Attribute Value**. The Attribute Type is a 2 octet field and has this structure:



- **Optional** - **0** is for Optional and **1** for Well-known.
- **Transitive** - **0** is for Transitive and **1** for Non-transitive.
- **Partial** - **0** is for when the Transitive attribute is partial and **1** is for when it is complete.
- **Extended Length** - **0** indicates that the attribute length is 1 octet and **1** indicates that it is 2 octets.
- **Attribute Type** - the codes for the attribute types are listed as follows:

1. ORIGIN
2. AS_PATH
3. NEXT_HOP
4. MULTI_EXIT_DISC
5. LOCAL_PREF
6. ATOMIC_AGGREGATE
7. AGGREGATOR
8. COMMUNITY
9. ORIGINATOR_ID
10. CLUSTER_LIST

We have already mentioned AS_PATH, however there are other Path Attributes that contribute to complex policy making being available in BGP-4. BGP router sends **Path Attributes** in the Update messages and these act as metrics. These Path Attributes apply to the destination networks, the BGP routes. These attributes fall into a number of categories:

- **Well-known mandatory attributes** - must be included in updates propagated to all peers and includes **AS_PATH**, **NEXT-HOP** and **ORIGIN**.
- **Well-known Discretionary attributes** - includes **LOCAL_PREF** and **ATOMIC_AGGREGATE** and are optional attributes to include in updates.
- **Optional Transitive Attributes** - includes **AGGREGATOR** and **COMMUNITY** and should be accepted by BGP even if the attribute is not supported by that router, it should pass on the attribute.
- **Optional Non-transitive Attribute** - Includes the **MULTI_EXIT_DISC (MED**, the **ORIGINATOR_ID** and **CLUSTER_LIST**. Non-transitive means that if the BGP router does not recognise the attribute it can ignore it and not pass it on.

A transitive attribute that is not implemented by a router can be passed on to another router and is called 'partial'. A non-transitive attribute has to be deleted by a router if it hasn't implemented it.

ORIGIN

The mandatory **Origin Attribute** specifies the origin of a routing update. It uses the letter **i** to indicate that the NLRI was learned from a protocol inside the originating AS i.e. an IGP (e.g. by using the **network** statement). The letter **e** indicates that the NLRI has been learned from EGP externally, and a **?** means that the NLRI is incomplete and was probably redistributed into BGP for which the source is obviously unknown. IGP is preferred over EGP which in turn is preferred over 'unknown'.

The Attribute Value Codes for these are **0** for AS_SET, **1** for EGP and **2** for incomplete.

AS_PATH

The path to the network specified by the NLRI is shared in the form of **Path Vectors** that contain AS numbers which a route should take to the destination network. Both **BGP-3** and **BGP-4** carry AS numbers of the ASs that have been traversed using the mandatory **AS_PATH** attribute and a router will reject updates containing its own AS number so preventing loops. When a BGP speaker originates a route it adds it's own AS number to the AS_PATH attribute for the NLRI sent in an update to an EBGP peer. Subsequent BGP routers **prepend** their own ASs so the AS_PATH attribute grows from the beginning i.e. the

originating AS is at the end of the string. This type of AS_PATH is strictly called an **AS_SEQUENCE**. Other types include **AS_SET**, **AS_CONFED_SEQUENCE** and **AS_CONFED_SET**. The Attribute Value Codes for these are **1** for AS_SET, **2** for AS_SEQUENCE, **3** for AS_CONFED_SET and **4** for AS_CONFED_SEQUENCE.

If there are multiples paths to a destination network, BGP prefers the route with the shortest AS_PATH. Manipulation of where packets will go can be carried out by a BGP router modifying the AS_PATH. For instance, there may be two paths to a particular network but the shortest AS_PATH takes packets down a slower link than the other path. The router can prepend any number of AS numbers to this AS_PATH to increase the length of the AS_PATH, thereby forcing packets down the longer but more favourable route. It is considered wise to just use multiple instances of the local AS number to increase the length of the AS_PATH rather than use arbitrary AS numbers in case of loops forming.

ATOMIC_AGGREGATE

A BGP router may advertise routes that 'overlap' meaning for example, both routes 20.1.0.0/16 and 20.1.1.0/24 may be advertised where 20.1.0.0/16 is an aggregate which includes the route 20.1.1.0/24. Routers always prefer the route with the longest mask (more specific route). When aggregating routes path information is lost as mentioned before, because the router advertising the aggregate just includes it's own AS in the AS_PATH rather than include the path information for the original routes being aggregated.

The **Atomic Aggregate** attribute informs a BGP neighbour that the router has aggregated the IP networks. As the route is advertised further to other peers the ATOMIC_AGGREGATE has to remain attached to the route.

Another optional attribute called the **AGGREGATOR** indicates the router ID (IP address) and the AS number of the router that performed the aggregation.

AS_SET

Where the AS_PATH atribute (AS_SEQUENCE) is an ordered list, the AS_SET is unordered.

Why have an unordered list? Well, when aggregation is performed the AS_PATH information is lost and the aggregate is advertised as originating from the BGP peer that performs the aggregation. The problem with this is that loop detection cannot occur and the aggregate may be advertised back to an AS which already knows about one or more of the specific routes included in the aggregate. Maintaining an ordered AS_PATH history for each of the specific routes is not possible, but an unordered list is perfectly possible and this is where AS_SET is useful.

The aggregating router has the AS_SEQUENCE information and just includes this as a jumbled list of ASs in the AS_SET attribute so helping other peers to still prevent routes. A route that has the AS_SET attribute need not have the ATOMIC_AGGREGATE.

NEXT_HOP

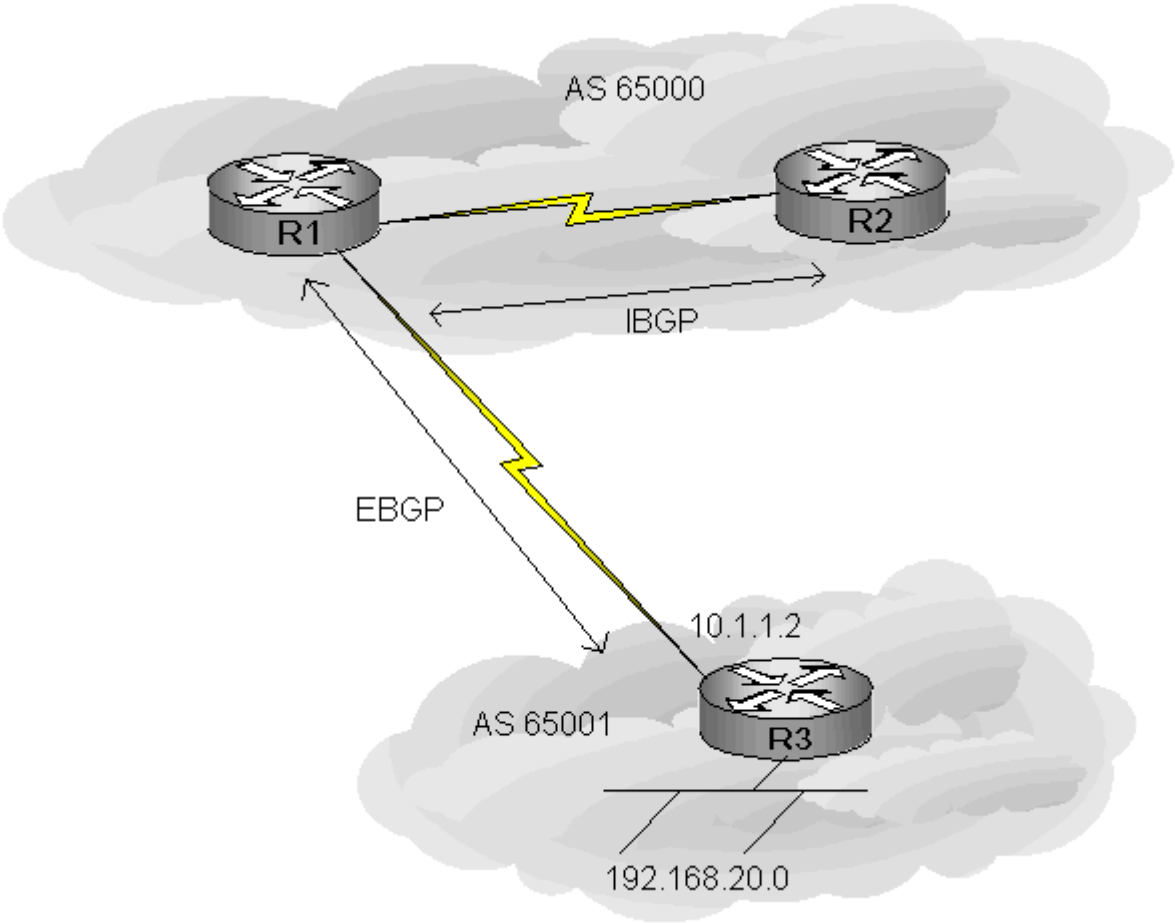
Consider these scenarios:

- For EBGP peers the next hop to an *external* destination network is the IP address of the EBGP peer that sent the update.
- For IBGP peers the next hop to an *internal* network is that of the IBGP neighbour that originated the route. If this neighbour is separated by an IGP (i.e. the IBGP peers do

not share a common link) then a recursive lookup must be performed to find the route to this next hop.

- Alternatively, for IBGP peers the next hop to an *external* network is the EBGP peer that learned the route, rather than the router that sent the update, since the IBGP router may not be the best located to get to the other AS. On a multi-access network like Ethernet, this is fine, since there is a route to the router in the other AS. There are however, issues if the network is a Non-broadcast Multi-access (NBMA) network such as Frame Relay.

Take the following example:



R1 and R3 are EBGP peers. R3 advertises the network 192.168.20.0 with the NEXT_HOP attribute set to 10.1.1.2 which is part of AS65001 NOT AS65000! R2 and R1 have an IBGP peering, however although R1 can advertise the 192.168.20.0 network to R2, when R2 performs a recursive route lookup it will not find the route to 10.1.1.0 network since it is not in AS65000. The route 192.168.20.0 will be in the BGP table but NOT installed as a BGP route in the IGP routing table because it is unreachable. You could get around this by setting a static route, or run the IGP in passive (listening) mode or set the AS border router (R1) as the next hop using the **next-hop-self** switch. This next hop address is known to the IGP.

LOCAL_PREF

The **Local Preference** is relevant when there is more than one path to a network outside of the current AS for instance if your network is connected to more than one ISP. Each of the routers that link to outside the AS can set a preference value for routes advertised into the AS, and this value indicates the router's preference for these routes. Only IBGP routers share the

local preference values it does not leave the AS. The higher the value the more preferable the route is so if there are multiple paths to this network the route with the highest Local Preference is chosen and all traffic destined for the network is sent this way.

WEIGHT

The **Weight** attribute is specific to Cisco. Router-originated routes have a weight of 32768 by default and other routes have a weight of zero where higher weight routes are preferred. Weight acts in the same way as Local Preference, the only difference is that it only applies to routes within the box and is not communicated to other peers. If two peers are advertising the same route to a particular peer, then that peer can assign a higher weight to routes learned from one of those peers, and these routes would be preferred.

MULTI_EXIT_DISC

The **Multi-Exit-Discriminator (MED)** (BGP-2 and BGP-3 called this the INTER_AS metric) is used between EBGP peers when there are multiple paths from one AS to another and it indicates to external neighbours which path is preferred into an AS. The reason may be that one link has a higher capacity than another link. As with metrics the lower the MED the more preferable the path is. The LOCAL_PREF influences traffic leaving an AS whilst the MED influences traffic entering an AS.

The MED attribute is only used between directly connected ASs so it is not passed onwards to other ASs, if MED attributes are required there then they would have to be set separately.

COMMUNITY

The **Community** attribute (or **Tag**) allows BGP communities to be set up and provides a way of grouping destinations according to common BGP attributes, filters and policies.

The Community attribute is made up of four octets where the first two indicate the AS and the last two contain the community identifier. Cisco puts these the other way around by default, although this can be changed. If a route from AS 50 has the identifier 30 then the Community attribute will be 0x0032001e where 0x0032 is 50 and 0x001e is 30.

Reserved Community attributes are 0x00000000 to 0x0000ffff and 0xffff0000 to 0xffffffff. From these reserved numbers there are four well-known communities:

- **INTERNET** - by default all routes belong to this community and are advertised.
- **NO_EXPORT** - this has the value 0xffffff01 and routes with this attribute cannot be advertised to EBGP peers i.e. outside of their AS with the exception of internal ASs within a Confederation.
- **NO_ADVERTISE** - this has the value 0xffffff02 and routes with this attribute cannot be advertised to either EBGP or IBGP peers.
- **LOCAL_AS** - this has the value 0xffffff03 and acts in the same way as NO_EXPORT except that routes with this attribute cannot even be advertised between EBGP peers in private ASs within a Conferation

Cisco originally invented this attribute but it is now a standard and is detailed in RFC 1997.

Use of the Community attribute is described in RFC 1998. Communities are used to apply policies to a collection of routes. If a router sets a community attribute to a particular value for

a group of routes, then the neighbours can apply their filtering, redistribution and attribute change policies to a group of routes based on the Community attribute rather than one at a time. One route can be assign more than one Community attribute. An aggregate route inherits all the community attributes of all the routes that are being aggregated.

ORIGINATOR_ID

This attribute is 32 bits long and is used by a Route Reflector as a Reflector ID (RID) to ensure that no loops occur in an AS using Route Reflectors.

CLUSTER_LIST

This attribute lists the route reflector cluster IDs that the route has passed through so that if a route reflector sees it's own cluster ID it drops the route to stop loops.

Peer Groups

We have mentioned the Community attribute to ease administration of policies for groups of routes. You can also set up groups of peers rather than routes and apply policies to groups of routers rather than groups of routes. Routers then only need to consult the policy database once and send copies of updates to multiple routers rather than perform multiple lookups.

Decision Criteria in a Multi-homed Environment

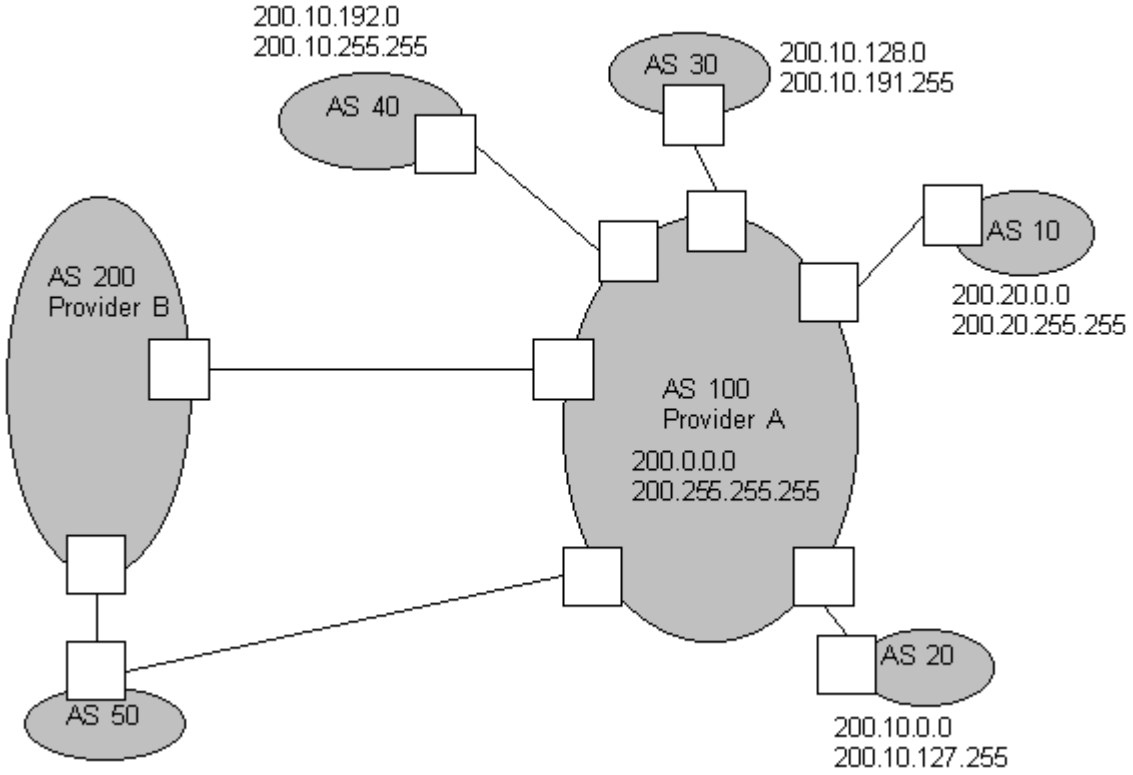
In a Multi-homed environment where resilience means that there are multiple connnections to ISPs BGP only chooses one route to a particular destination and uses the following priorities in order when making its decision:

1. Only look at synchronised routes with a valid next hop.
2. Prefer highest WEIGHT (Cisco only).
3. Prefer highest LOCAL_PREF.
4. Prefer route originated by the local router.
5. Prefer shortest AS_PATH.
6. Prefer IGP origin code over EGP and furthermore over incomplete.
7. Prefer lowest MED.
8. Prefer EBGP path rather than IBGP path.
9. Prefer the path through the closest IGP neighbour.
10. Prefer the oldest EBGP path.
11. Prefer the path with the lowest BGP router ID.

An Example demonstrating AS_PATH, AS_SET and the NLRI

In BGP terminology, a route is made up of **Network Layer Reachability Information (NLRI)** and path attributes. In BGP-4 the NLRI consists of an IP address prefix and a prefix length which is the number of bits that make up the range of addresses (like a subnet mask, but NOT), so a class B address would have 16-bit prefix length. BGP-4 does not respect the traditional class distinctions, e.g. 10.1.0.0 with a 16-bit prefix length would be treated like a class B address instead of a class A address as most other protocols would. This is the implementation of CIDR. BGP-4 can aggregate many networks into a single advertisement

similar to OSPF summaries and indicates this with the two Aggregate Attributes described earlier.



Provider A advertises a BGP NLRI of 200.0.0.0/8 to provider B so that B knows that A is taking care of hosts 200.0.0.0 to 200.255.255.255. 'A' administers AS10, AS20, AS30 and AS40 and these can use a default route to access the routing tables on any router in AS100.

AS200 learns two paths to 200.20.0.0, one AS path being (AS10, AS100) and the other being (AS10, AS100, AS50), the administrator can assign weights to these ASs in order to develop a policy on which path to use.

One rule in BGP is that a router in a particular AS will ignore routes that contain its own AS number since they represent a loop. For instance, when AS200 advertises to AS100 the paths it learns to 200.10.0.0, i.e. the AS path (AS100, AS20), then BGP in AS100 ignores it as it can see the AS100 in the AS path.

Another rule in BGP is that you should only advertise paths that you use. For example, routers running BGP in AS50 have two paths to 200.20.0.0, one being (AS10, AS100) and one from a router in AS200 (based on its choice of path).

You can use address aggregation in BGP-4 so that, instead of just advertising a single path you can advertise an AS set e.g. (10,20,30,40,100), which means any one of these ASs.

Although you can inject routes learned by BGP into ASs running an IGP (ideal for a stub AS), you need to create at least two peer BGP routers within an IGP AS (normally on the edge of the AS) in order to maintain AS_path information if the AS is a transit AS, this is because BGP attributes must be shared by all routers running BGP. Similarly, an external BGP session needs to be created to a router in a different AS.

Synchronisation

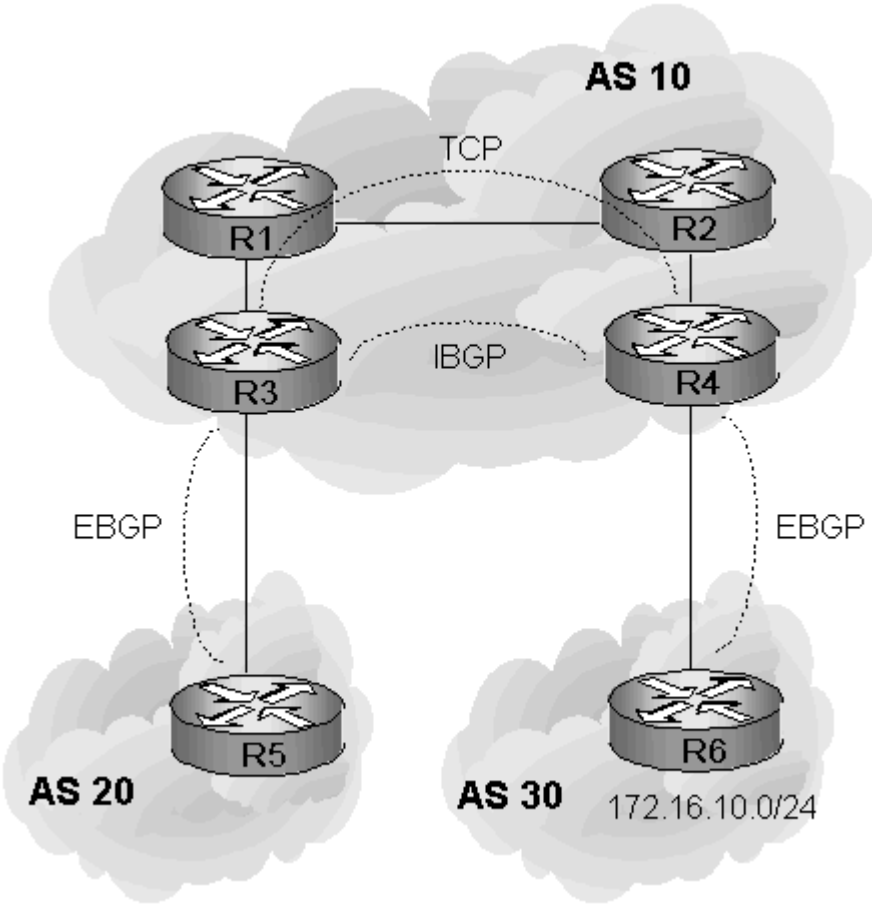
IBGP is used to pass routes learned from one EBGP (edge router) to another EBGP. Because IBGP peers are in the same AS, the AS_PATH does not change so there is no loop protection via the use of the AS_PATH attribute. The AS_PATH is only prepended when a

route is advertised to an EBGP peer. So an IBGP peer is not allowed to advertise routes that it learned about to another IBGP peer. An IBGP router can only advertise routes that it knows about (i.e. learned from an IGP) to its peers in other ASs i.e. EBGP peers.

The default configuration of BGP on a circuit does not advertise any routes or allow any learned routes into the IGP routing table, these have to be manually entered as Network statements or be redistributed into the IGP. The trouble with redistributing BGP routes into the IGP is that you can flood many thousands of routes into the IGP if you are not careful. This can overload some routers and bring down a network.

If you had a hub spoke topology of IBGP peers, the spokes will not share learned routes with each other. If you wanted to operate BGP with full reachability within an AS and prevent routing loops, you would need to configure a fully-meshed IBGP peering topology, if you were not redistributing BGP routes into the IGP.

Consider the following topology where the IBGP peers in AS 10 are partially meshed.



An IGP such as RIP or OSPF is used in AS 10 to provide TCP connectivity between R3 and R4. R3 and R4 are IBGP peers with each other, plus they have EBGP peerings with routers in AS 20 and AS 30 respectively. They therefore share routes learned from these different ASs with each other. The TCP connection for the IBGP peering however, is routed using the IGP through R1 and R2.

If R4 learns of the route 172.16.10.0/24 from R6 in AS 30 and advertises this to R3. R3 will advertise this route to AS 20. Any packets in AS 20 destined for 172.16.10.0/24, will now be forwarded to R3. R3 knows that to get to 172.16.10.0/24 it needs to get these packets to R4, however it reaches R4 via R1 and R2 using the IGP so packets destined for 172.16.10.0/24 are sent to R1. R1 however, knows nothing of the network 172.16.10.0/24 and so drops the packet thereby creating a black hole.

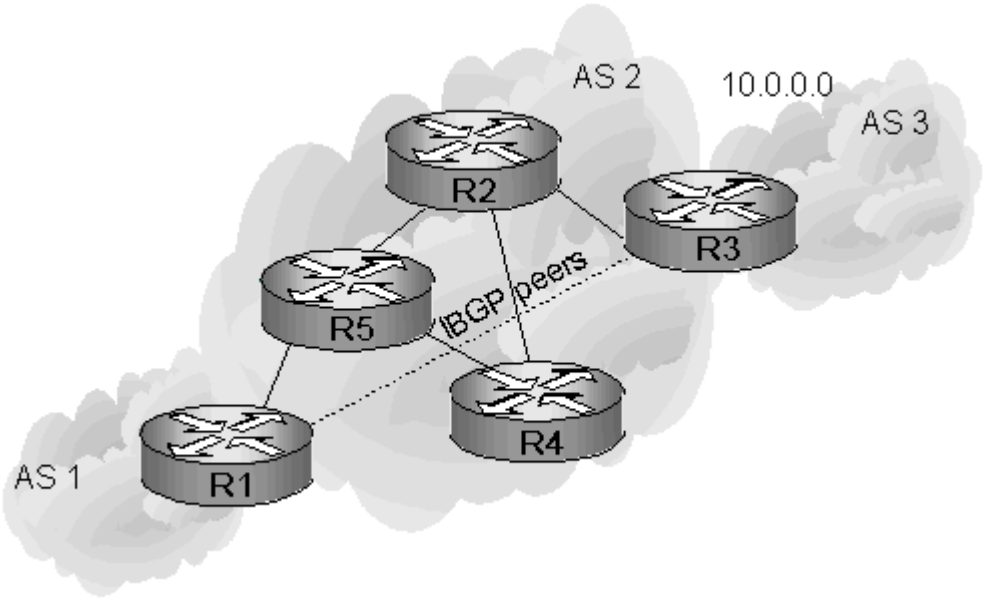
If the IGP knows about the same routes that BGP does then this black hole would not occur. The process of making sure that both BGP and the IGP know the same routes is called

Synchronisation. This rule states that **a route must first be known via IGP before this same route learned from an IBGP peer is advertised to any BGP peers or entered into the IGP routing table as a BGP route.**

This 'BGP Synchronisation' prevents **Black Holes** and creates consistency of routing information throughout the AS. It cannot however, influence how the other AS will route traffic and it assumes that you are redistributing routes between the IGP and BGP.

In a fully meshed BGP network without an IGP, all routers can learn all the routes without any gaps, but if synchronisation is still operating as discussed earlier, no BGP learned routes will be entered into the routing table. In this fully meshed scenario, synchronisation can be turned off and routes learned by one peer will be allowed to be advertised and to be entered into the IGP routing table. The fully meshed BGP network is becoming the way forward for many ISPs.

Take another example where there is no redistribution between the IGP and BGP:



R1 and R3 are IBGP peers and R1 knows how to get to the 10.0.0.0. R2, R4 and R5 however do not know about the 10.0.0.0 network, all they see are IP packets being routed through them hop by hop. If R2, R4 and R5 need to know about the 10.0.0.0 network then you need to redistribute between the BGP and the IGP and have synchronisation on so that routes are not advertised into BGP unless it exists in the IGP routing table. Or you can have IBGP fully meshed peers everywhere and turn off Synchronisation. The latter option is impractical in even moderately sized networks, which is where techniques such as Router Reflectors and Confederations are useful.

Route Reflectors

As discussed earlier, in order to prevent routing loops the normal BGP rule is that routes learned via IBGP are not propagated to other IBGP peers, sometimes called **BGP Split Horizon**. All BGP Speakers have to be fully meshed in TCP/IP. This can be very tedious to set up plus impossible for more than a few routers, so the idea of **Route Reflectors** was introduced whereby a BGP Speaker would sit at the hub of a hub-spoke arrangement and become a Route Reflector for the spoke BGP Speakers. In contrast to the normal rules, this Route Reflector *can* propagate routes learned via IBGP to other IBGP peers and does not affect the paths that normal packets travel along. The Route Reflector 'reflects' routes learned via one IBGP peer in the cluster to the other IBGP peers in the cluster. The peers that are not

Route Reflectors are sometimes called Route Reflector clients, other BGP routers in the AS that are not part of the cluster are called 'non-clients'.

The rules for when a reflector updates are as follows:

- An update from a client peer is sent to all client and non-client peers.
- An update from a non-client peer is sent to all client peers.
- An update from a EBGP peer is sent to all client and non-client peers.

Route Reflectors are defined in RFC 1966.

There can be multiple Router Reflectors in the same cluster and at different levels (nested). A Route Reflector forms a peering with the other non-route reflectors called **Clients** and forms a **Cluster** with them. The clients are not peers with each other. Other IBGP routers not in the Cluster are called **Non-clients**. The **ORIGINATOR_ID** attribute is used to identify the router-id of the route reflector this enables the router to determine whether the route has come from itself in the first place thereby preventing loops. Another way of preventing loops is by use of cluster lists. If you want multiple route reflectors for resilience then a **CLUSTER_LIST** attribute can be configured so that the route reflectors recognise each other as being part of the same cluster. You can have multiple clusters as well and a Route Reflector can determine whether a loop exists by looking for its own cluster ID in the Cluster List of the advertisement. It is important that the route reflectors themselves are fully meshed.

Confederations

In very large BGP Autonomous Systems with fully-meshed IBGP peers you can divide up the routers into smaller private (member) ASs and form a confederation of private ASs that come together to produce a public AS as far as the external networks are concerned. Confederations are described in RFC 1965.

Reserved AS numbers used for private ASs are 64512 through to 65535. These numbers should therefore be used for the internal AS numbers. EBGP peers in each private AS peer with each other and the whole confederation has a **Confederation ID** which is a legitimate AS number that external ASs see. The confederation structure itself is invisible to the external ASs. Other ASs connect using the Confederation AS number. Normally Next-hop, metric and Local Preference information is not passed between EBGP peers, however within a confederation this information is shared.

In order to prevent loops from occurring the AS_PATH has two versions, one called the **AS_CONFED_SEQUENCE** that is an ordered list of ASs that are internal to the confederation, and the other called the **AS_CONFED_SET** that is an unordered list of ASs.

In large ASs Confederations can be used in conjunction with Route Reflectors.

RFCs

Other RFCs on BGP include:

RFC 1403 - BGP interaction with OSPF.
RFC 1773 - Experience of BGP-4.
RFC 1774 - Protocol Analysis
RFC 1863 - BGP/IDRP Route Server
RFC 2042 - Registering new attributes
RFC 2283 - Multiprotocol Extensions for BGP-4

RFC 2385 - BGP and TCP MD5 signatures
RFC 2439 - BGP Route Flap Damping