Border Gateway Protocol (**BGP**) is a standardized exterior gateway protocol designed to exchange routing and reachability information among autonomous systems (AS) on the Internet.[1] BGP is classified as a path- vector routing protocol,[2] and it makes routing decisions based on paths, network policies, or rule-sets configured by a network administrator.

BGP used for routing within an autonomous system is called **Interior Border Gateway Protocol**, **Internal BGP** (**iBGP**). In contrast, the Internet application of the protocol is called **Exterior Border Gateway Protocol**, **External BGP** (**eBGP**).

The Border Gateway Protocol was first described in 1989 in RFC 1105, and has been in use on the Internet since 1994.[3] IPv6 BGP was first defined in RFC 1883 in 1995, and it was improved to RFC 2283 in 1998.

The current version of BGP is version 4 (BGP4), which was published as RFC 4271 in 2006.[4] RFC 4271 corrected errors, clarified ambiguities and updated the specification with common industry practices. The major enhancement was the support for Classless Inter-Domain Routing (CIDR) and use of route aggregation to decrease the size of routing tables. The new RFC allows BGP4 to carry a wide range of IPv4 and IPv6 "address families". It is also called the Multiprotocol Extensions which is Multiprotocol BGP (MP-BGP).

BGP neighbors, called peers, are established by manual configuration among routers to create a TCP session on port 179. A BGP speaker sends 19-byte keep-alive messages every 60 seconds[5] to maintain the connection.[6] Among routing protocols, BGP is unique in using TCP as its transport protocol.

When BGP runs between two peers in the same autonomous system (AS), it is referred to as *Internal BGP* (*i- BGP* or *Interior Border Gateway Protocol*). When it runs between different autonomous systems, it is called *External BGP* (*eBGP* or *Exterior Border Gateway Protocol*). Routers on the boundary of one AS exchanging information with another AS are called *border* or *edge routers* or simply *eBGP peers* and are typically connected directly, while *i-BGP peers* can be interconnected through other intermediate routers. Other deployment topologies are also possible, such as running eBGP peering inside a VPN tunnel, allowing two remote sites to exchange routing information in a secure and isolated manner.

The main difference between iBGP and eBGP peering is in the way routes that were received from one peer are propagated to other peers. For instance, new routes learned from an eBGP peer are typically redistributed to all iBGP peers as well as all other eBGP peers (if *transit* mode is enabled on the router). However, if new routes are learned on an iBGP peering, then they are re-advertised only to all eBGP peers. These route- propagation rules effectively require that all iBGP peers inside an AS are interconnected in a full mesh.

How routes are propagated can be controlled in detail via the *route-maps* mechanism. This mechanism consists of a set of rules. Each rule describes, for routes matching some given criteria, what action should be taken. The action could be to drop the route, or it could be to modify some attributes of the route before inserting it in the routing table.

During the peering handshake, when OPEN messages are exchanged, BGP speakers can negotiate optional capabilities of the session,[7] including multiprotocol extensions[8] and various recovery modes. If the multiprotocol extensions to BGP are negotiated at the time of creation, the BGP speaker can prefix the Network Layer Reachability Information (NLRI) it advertises with an address family prefix. These families include the IPv4 (default), IPv6, IPv4/IPv6 Virtual Private Networks and multicast BGP. Increasingly, BGP is used as a generalized signaling protocol to carry information about routes that may not be part of the global Internet, such as VPNs.[9]

In order to make decisions in its operations with peers, a BGP peer uses a simple finite state machine (FSM) that consists of six states: Idle; Connect; Active; OpenSent; OpenConfirm; and Established. For each peer-to- peer session, a BGP implementation maintains a state variable that tracks which of these six states the session is in. The BGP defines the messages that each peer should exchange in order to change the session from one state to another.

The first state is the Idle state. In the Idle state, BGP initializes all resources, refuses all inbound BGP connection attempts and initiates a TCP connection to the peer. The second state is Connect. In the Connect state, the router waits for the TCP connection to complete and transitions to the OpenSent state if successful. If unsuccessful, it starts the ConnectRetry timer and transitions to the Active state upon expiration. In the Active state, the router resets the ConnectRetry timer to zero and returns to the Connect state. In the OpenSent state, the router sends an Open message and waits for one in return in order to transition to the OpenConfirm state. Keepalive messages are exchanged and, upon successful receipt, the router is placed into the Established state. In the Established state, the router can send and receive: Keepalive; Update; and Notification messages to and from its peer.

In the simplest arrangement, all routers within a single AS and participating in BGP routing must be configured in a full mesh: each router must be configured as a peer to every other router. This causes scaling problems, since the number of required connections grows quadratically with the number of routers involved. To alleviate the problem, BGP implements two options: route reflectors (RFC 4456) and BGP confederations (RFC 5065). The following discussion of basic UPDATE processing assumes a full iBGP mesh.

A given BGP router may accept Network Layer Reachability Information (NLRI) UPDATEs from multiple neighbors and advertise NLRI to the same, or a different set, of neighbors. Conceptually, BGP maintains its own master routing table, called the *local routing information base* (Loc-RIB), separate from the main routing table of the router. For each neighbor, the BGP process maintains a conceptual *adjacent routing information base, incoming* (Adj-RIB-In) containing the NLRI received from the neighbor, and a conceptual outgoing information base (Adj-RIB-Out) for NLRI to be sent to the neighbor.

The physical storage and structure of these conceptual tables are decided by the implementer of the BGP code. Their structure is not visible to other BGP routers, although they usually can be interrogated with management commands on the local router. It is quite common, for example, to store the two Adj-RIBs and the Loc-RIB together in the same data structure, with additional information attached to the RIB entries. The additional information tells the BGP process such things as whether individual entries belong in the Adj-RIBs for specific neighbors, whether the peer-neighbor route selection process made received policies eligible for the Loc-RIB, and whether Loc-RIB entries are eligible to be submitted to the local router's routing table management process.

BGP will submit the routes that it considers best to the main routing table process. Depending on the implementation of that process, the BGP route is not necessarily selected. For example, a directly connected prefix, learned from the router's own hardware, is usually most preferred. As long as that directly connected route's interface is active, the BGP route to the destination will not be put into the routing table. Once the interface goes down, and there are no more preferred routes, the Loc-RIB route would be installed in the main routing table.

Until recently, it was a common mistake to say, "BGP carries policies." BGP actually carried the information with which rules inside BGP-speaking routers could make policy decisions. Some of the information carried that is explicitly intended to be used in policy decisions are communities and multi-exit discriminators (MED).

The BGP standard specifies a number of decision factors, more than the ones that are used by any other common routing process, for selecting NLRI to go into the Loc-RIB. The first decision point for evaluating NLRI is that its next-hop attribute must be reachable (or resolvable). Another way of saying the next-hop must be reachable is that there must be an active route, already in the main routing table of the router, to the prefix in which the next-hop address is reachable.

Next, for each neighbor, the BGP process applies various standard and implementation-dependent criteria to decide which routes conceptually should go into the Adj-RIB-In. The neighbor could send several possible routes to a destination, but the first level of preference is at the neighbor level. Only one route to each destination will be installed in the conceptual Adj-RIB-In. This process will also delete, from the Adj-RIB-In, any routes that are withdrawn by the neighbor.

Whenever a conceptual Adj-RIB-In changes, the main BGP process decides if any of the neighbor's new routes are preferred to routes already in the Loc-RIB. If so, it replaces them. If a given route is withdrawn by a neighbor, and there is no other route to that destination, the route is removed from the Loc-RIB and no longer sent by BGP to the main routing table manager. If the router does not have a route to that destination from any non-BGP source, the withdrawn route will be removed from the main routing table.

After verifying that the next hop is reachable, if the route comes from an internal (i.e. iBGP) peer, the first rule to apply, according to the standard, is to examine the

LOCAL_PREFERENCE attribute. If there are several iBGP routes from the neighbor, the one with the highest LOCAL_PREFERENCE is selected unless there are several routes with the same LOCAL_PREFERENCE. In the latter case the route selection process moves to the next tiebreaker. While LOCAL_PREFERENCE is the first rule in the standard, once reachability of the NEXT_HOP is verified, Cisco and several other vendors first consider a decision factor called WEIGHT which is local to the router (i.e. not transmitted by BGP). The route with the highest WEIGHT is preferred.

The LOCAL_PREFERENCE, WEIGHT, and other criteria can be manipulated by local configuration and software capabilities. Such manipulation, although commonly used, is outside the scope of the standard. For example, the COMMUNITY attribute (see below) is not directly used by the BGP selection process. The BGP neighbor process however can have a rule to set LOCAL_PREFERENCE or another factor based on a manually programmed rule to set the attribute if the COMMUNITY value matches some pattern matching criterion. If the route was learned from an external peer the per-neighbor BGP process computes a LOCAL_PREFERENCE value from local policy rules and then compares the LOCAL_PREFERENCE of all routes from the neighbor.

At the per-neighbor level – ignoring implementation-specific policy modifiers – the order of tie-breaking rules is:

1. Prefer the route with the shortest AS_PATH. An AS_PATH is the set of AS numbers that must be traversed to reach the advertised destination. AS1-AS2-AS3 is shorter than AS4-AS5-AS6- AS7.
2. Prefer routes with the lowest value of their ORIGIN attribute.
3. Prefer routes with the lowest MULTI_EXIT_DISC (multi-exit discriminator or MED) value.[a]

Once candidate routes are received from neighbors, the Loc-RIB software applies additional tie-breakers to routes to the same destination.

1. If at least one route was learned from an external neighbor (i.e., the route was learned from eBGP), drop all routes learned from iBGP.
2. Prefer the route with the lowest interior cost to the NEXT_HOP, according to the main routing table. If two neighbors advertised the same route, but one neighbor is reachable via a low- bitrate link and the other by a high-bitrate link, and the interior routing protocol calculates lowest cost based on highest bitrate, the route through the high-bitrate link would be preferred and other routes dropped.

   If there is more than one route still tied at this point, several BGP implementations offer a configurable option to load-share among the routes, accepting all (or all up to some number).

3. Prefer the route learned from the BGP speaker with the numerically lowest BGP identifier

4. Prefer the route learned from the BGP speaker with the lowest peer IP address

BGP communities are attribute tags that can be applied to incoming or outgoing prefixes to achieve some common goal (RFC 1997). While it is common to say that BGP allows an administrator to set policies on how prefixes are handled by ISPs, this is generally not possible, strictly speaking. For instance, BGP natively has no concept to allow one AS to tell another AS to restrict advertisement of a prefix to only North American peering customers. Instead, an ISP generally publishes a list of well-known or proprietary communities with a description for each one, which essentially becomes an agreement of how prefixes are to be treated. RFC 1997 also defines three well-known communities that have global significance; NO_EXPORT, NO_ADVERTISE and NO_EXPORT_SUBCONFED. RFC 7611 defines ACCEPT_OWN. Examples of common communities include local preference adjustments, geographic or peer type restrictions, DoS avoidance (black holing), and AS prepending options. An ISP might state that any routes received from customers with community XXX:500 will be advertised to all peers (default) while community XXX:501 will restrict advertisement to North America. The customer simply adjusts their configuration to include the correct community or communities for each route, and the ISP is responsible for controlling who the prefix is advertised to. The end user has no technical ability to enforce correct actions being taken by the ISP, though problems in this area are generally rare and accidental.

It is a common tactic for end customers to use BGP communities (usually ASN:70,80,90,100) to control the local preference the ISP assigns to advertised routes instead of using MED (the effect is similar). The community attribute is transitive, but communities applied by the customer very rarely become propagated outside the next-hop AS. Not all ISPs give out their communities to the public, while some others do.[10]

The BGP Extended Community Attribute was added in 2006, in order to extend the range of such attributes and to provide a community attribute structuring by means of a type field. The extended format consists of one or two octets for the type field followed by seven or six octets for the respective community attribute content. The definition of this Extended Community Attribute is documented in RFC 4360. The IANA administers the registry for BGP Extended Communities Types.[11] The Extended Communities Attribute itself is a transitive optional BGP attribute. However, a bit in the type field within the attribute decides whether the encoded

extended community is of a transitive or non-transitive nature. The IANA registry therefore provides different number ranges for the attribute types. Due to the extended attribute range, its usage can be manifold. RFC 4360 exemplarily defines the "Two-Octet AS Specific Extended Community", the "IPv4 Address Specific Extended Community", the "Opaque Extended Community", the "Route Target Community", and the "Route Origin Community". A number of BGP QoS drafts[12] also use this Extended Community Attribute structure for inter-domain QoS signalling.

Note: since RFC 7153, extended communities are compatible with 32-bit ASNs.

With the introduction of 32-bit AS numbers, some issues were immediately obvious with the community attribute that only defines a 16 bits ASN field, which prevents the matching between this field and the real ASN value. It is the reason why RFC 8092 and RFC 8195 introduce a Large Community (http://largebgpcom munities.net/) attribute of 12 bytes, divided in three field of 4 bytes each (AS:function:parameter).

MEDs, defined in the main BGP standard, were originally intended to show to another neighbor AS the advertising AS's preference as to which of several links are preferred for inbound traffic. Another application of MEDs is to advertise the value, typically based on delay, of multiple AS that have presence at an IXP, that they impose to send traffic to some destination.

BGP is "the most scalable of all routing protocols."[13]

An autonomous system with internal BGP (iBGP) must have all of its iBGP peers connect to each other in a full mesh (where everyone speaks to everyone directly). This full-mesh configuration requires that each router maintain a session to every other router. In large networks, this number of sessions may degrade performance of routers, due to either a lack of memory, or high CPU process requirements.

Route reflectors[14] reduce the number of connections required in an AS. A single router (or two for redundancy) can be made a route reflector: other routers in the AS need only be configured as peers to them. A route reflector offers an alternative to the logical full-mesh requirement of internal border gateway protocol (IBGP). A RR acts as a for IBGP sessions. The purpose of the RR is concentration. Multiple BGP routers can peer with a central point, the RR – acting as a route reflector server – rather than peer with every other router in a full mesh. All the other IBGP routers become route reflector clients.

This approach, similar to OSPF's DR/BDR feature, provides large networks with added IBGP scalability. In a fully meshed IBGP network of 10 routers, 90 individual CLI statements (spread throughout all routers in the topology) are needed just to define the remote-AS of each peer: this quickly becomes a headache to manage. A RR topology could cut these 90 statements down to 18, offering a viable solution for the larger networks administered by ISPs.

A route reflector is a single point of failure, therefore at least a second route reflector may be configured in order to provide redundancy. As it is an additional peer for the other 10 routers, it comes with the additional statement count to double that minus 2 of the single Route Reflector setup. An additional 11*2-2=20 statements in this case due to adding the additional Router. Additionally, in a BGP multipath Environment this also can benefit by adding local switching/Routing throughput if the RRs are acting as traditional Routers instead of just a dedicated Route Reflector Server role.

RR servers propagate routes inside the AS based on the following rules:

If a route is received from a non-client peer, reflect to clients only and EBGP peers.

If a route is received from a client peer, reflect to all non-client peers and also to client peers, except the originator of the route and reflect to EBGP peers.

RR and its clients form a "Cluster". The "Cluster-ID" is then attached to every route advertised by RR to its client or nonclient peers. Cluster-ID is a cumulative, non-transitive BGP attribute and every RR MUST prepend the local CLUSTER_ID to the CLUSTER_LIST in order to avoid routing loops. Route reflectors and confederations both reduce the number of iBGP peers to each router and thus reduce processing overhead. Route reflectors are a pure performance-enhancing technique, while confederations also can be used to implement more fine-grained policy.

Confederations are sets of autonomous systems. In common practice,[15] only one of the confederation AS numbers is seen by the Internet as a whole. Confederations are used in very large networks where a large AS can be configured to encompass smaller more manageable internal ASs.

The confederated AS is composed of multiple ASs. Each confederated AS alone has iBGP fully meshed and has connections to other ASs inside the confederation. Even though these ASs have eBGP peers to ASs within the confederation, the ASs exchange routing as if they used iBGP. In this way, the confederation preserves next hop, metric, and local preference information. To the outside world, the confederation appears to be a single AS. With this solution, iBGP transit AS problems can be resolved as iBGP requires a full mesh between all BGP routers: large number of TCP sessions and unnecessary duplication of routing traffic.

Confederations can be used in conjunction with route reflectors. Both confederations and route reflectors can be subject to persistent oscillation unless specific design rules, affecting both BGP and the interior routing protocol, are followed.[16]

However, these alternatives can introduce problems of their own, including the following: route oscillation, sub-optimal routing, increase of BGP convergence time.

Additionally, route reflectors and BGP confederations were not designed to ease BGP router configuration. Nevertheless, these are common tools for experienced BGP network architects. These tools may be combined, for example, as a hierarchy of route reflectors.

The routing tables managed by a BGP implementation are adjusted continually to reflect actual changes in the network, such as links breaking and being restored or routers going down and coming back up. In the network as a whole it is normal for these changes to happen almost continuously, but for any particular router or link, changes are supposed to be relatively infrequent. If a router is misconfigured or mismanaged then it may get into a rapid cycle between down and up states. This pattern of repeated withdrawal and re-announcement known as route flapping can cause excessive activity in all the other routers that know about the broken link, as the same route is continually injected and withdrawn from the routing tables. The BGP design is such that delivery of traffic may not function while routes are being updated. On the Internet, a BGP routing change may cause outages for several minutes.

A feature known as *route flap damping* (RFC 2439 (http://www.ietf.org/rfc/rfc2439.txt)) is built into many BGP implementations in an attempt to mitigate the effects of route flapping. Without damping, the excessive activity can cause a heavy processing load on routers, which may in turn delay updates on other routes, and so affect overall routing stability. With damping, a route's flapping is exponentially decayed. At the first instance when a route becomes unavailable and quickly reappears, damping does not take effect, so as to maintain the normal fail-over times of BGP. At the second occurrence, BGP shuns that prefix for a certain length of time; subsequent occurrences are timed out exponentially. After the abnormalities have ceased and a suitable length of time has passed for the offending route, prefixes can be reinstated and its slate wiped clean. Damping can also mitigate denial of service attacks; damping timings are highly customizable.

It is also suggested in RFC 2439 (under "Design Choices -> Stability Sensitive Suppression of Route Advertisement") that route flap damping is a feature more desirable if implemented to Exterior Border Gateway Protocol Sessions (eBGP sessions or simply called exterior peers) and not on Interior Border Gateway Protocol Sessions (iBGP sessions or simply called internal peers); With this approach when a route flaps inside an autonomous system, it is not propagated to the external ASs – flapping a route to an eBGP will have a chain of flapping for the particular route throughout the backbone. This method also successfully avoids the overhead of route flap damping for iBGP sessions.

However, subsequent research has shown that flap damping can actually lengthen convergence times in some cases, and can cause interruptions in connectivity even when links are not flapping.[18][19] Moreover, as backbone links and router processors have become faster, some network architects have suggested that flap damping may not be as important as it used to be, since changes to the routing table can be handled much faster by routers.[20] This has led the RIPE Routing Working Group to write that "with the current implementations of BGP flap damping, the application of flap damping in ISP networks is NOT recommended. ... 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 ... . These side-effects would quite likely be worse than the impact caused by simply not running flap damping at all."[21] Improving stability without the problems of flap damping is the subject of current research.[22]

One of the largest problems faced by BGP, and indeed the Internet infrastructure as a whole, is the growth of the Internet routing table. If the global routing table grows to the point where some older, less capable routers cannot cope with the memory requirements or the CPU load of maintaining the table, these routers will cease to be effective gateways between the parts of the Internet they connect. In addition, and perhaps even more importantly, larger routing tables take longer to stabilize (see above) after a major connectivity change, leaving network service unreliable, or even unavailable, in the interim.

Until late 2001, the global routing table was growing exponentially, threatening an eventual widespread breakdown of connectivity. In an attempt to prevent this, ISPs cooperated in keeping the global routing table as small as possible, by using Classless Inter-Domain Routing (CIDR) and route aggregation. While this slowed the growth of the routing table to a linear process for several years, with the expanded demand for multihoming by end user networks the growth was once again superlinear by the middle of 2004.

A Y2K-like overflow triggered in 2014 for those models that were not appropriately updated.

While a full IPv4 BGP table as of August 2014 (512k day)[23][24] was in excess of 512,000 prefixes,[25] many older routers had a limit of 512k (512,000–524,288)[26][27] routing table entries. On August 12, 2014, outages resulting from full tables hit eBay, LastPass and Microsoft Azure among others.[28] A number of Cisco routers commonly in use had TCAM, a form of high-speed content-addressable memory, for storing BGP advertised routes. On impacted routers, the TCAM was default allocated as 512k IPv4 routes and 256k IPv6 routes. While the reported number of IPv6 advertised routes was only about 20k, the number of advertised IPv4 routes reached the default limit, causing a spillover effect as routers attempted to compensate for the issue by using slow software routing (as opposed to fast hardware routing via TCAM). The main method for dealing with this issue involves operators changing the TCAM allocation to allow more IPv4 entries, by reallocating some of the TCAM reserved for IPv6 routes, which requires a reboot on most routers. The 512k problem was predicted by a number of IT professionals. The actual allocations which pushed the number of routes above 512k was the announcement of about 15,000 new routes in short order, starting at 07:48 UTC. Almost all of these routes were to Verizon Autonomous Systems 701 and 705, created as a result of deaggregation of larger blocks, introducing thousands of new */24* routes, and making the routing table reach 515,000 entries. The new routes appear to have been reaggregated within 5 minutes, but instability across the Internet apparently continued for a number of hours.[32] Even if Verizon had not caused the routing table to exceed 512k entries in the short spike, it would have happened soon anyway through natural growth.

Route summarization is often used to improve aggregation of the BGP global routing table, thereby reducing the necessary table size in routers of an AS. Consider AS1 has been allocated the big address space of *172.16.0.0/16*, this would be counted as one route in the table, but due to customer requirement or traffic engineering purposes, AS1 wants to announce smaller, more specific routes of *172.16.0.0 /18* , *172.16.64.0/18*, and *172.16.128.0/18*. The prefix *172.16.192.0/18* does not have any hosts so AS1 does not announce a specific route *172.16.192.0/18*. This all counts as AS1 announcing four routes.

AS2 will see the four routes from AS1 (*172.16.0.0/16*, *172.16.0.0/18*, *172.16.64.0/18*, and *172.16.128.0/18*) and it is up to the routing policy of AS2 to decide whether or not to take a copy of the four routes or, as *172.16.0.0/16* overlaps all the other specific routes, to just store the summary, *172.16.0.0/16*.

If AS2 wants to send data to prefix *172.16.192.0/18*, it will be sent to the routers of AS1 on route *172.16.0.0/16*. At AS1's router, it will either be dropped or a destination unreachable ICMP message will be sent back, depending on the configuration of AS1's routers.

If AS1 later decides to drop the route *172.16.0.0/16*, leaving *172.16.0.0/18*, *172.16.64.0/18*, and *172.16.128.0/18*, AS1 will drop the number of routes it announces to three. AS2 will see the three routes, and depending on the routing policy of AS2, it will store a copy of the three routes, or aggregate the prefix's *172.16.0.0/18* and *172.16.64.0/18* to *172.16.0.0/17*, thereby reducing the number of routes AS2 stores to only two: *172.16.0.0/17* and *172.16.128.0/18*.

If AS2 wants to send data to prefix *172.16.192.0/18*, it will be dropped or a destination unreachable ICMP message will be sent back at the routers of AS2 (not AS1 as before), because *172.16.192.0/18* would not be in the routing table.

The RFC 1771 (*A Border Gateway Protocol 4 (BGP-4)*) planned the coding of AS numbers on 16 bits, for 64510 possible public AS, since ASN 64512 to 65534 were reserved for private use (0 and 65535 being forbidden). In 2011, only 15000 AS numbers were still available, and projections[33] were envisioning a complete depletion of available AS numbers in September 2013.

RFC 6793 extends AS coding from 16 to 32 bits (keeping the 16 bits AS range 0 to 65535, and its reserved AS numbers), which now allows up to 4 billion available AS. An additional private AS range is also defined in RFC 6996 (from 4200000000 to 4294967294, 4294967295 being forbidden by RFC 7300).

To allow the traversal of router groups not able to manage those new ASNs, the new attribute OT AS4_PATH is used.

32-bit ASN assignments started in 2007.

Another factor causing this growth of the routing table is the need for load balancing of multi-homed networks. It is not a trivial task to balance the inbound traffic to a multi-homed network across its multiple inbound paths, due to limitation of the BGP route selection process. For a multi-homed network, if it announces the same network blocks across all of its BGP peers, the result may be that one or several of its inbound links become congested while the other links remain under-utilized, because external networks all picked that set of congested paths as optimal. Like most other routing protocols, BGP does not detect congestion.

To work around this problem, BGP administrators of that multihomed network may divide a large contiguous IP address block into smaller blocks and tweak the route announcement to make different blocks look optimal on different paths, so that external networks will choose a different path to reach different blocks of that multi- homed network. Such cases will increase the number of routes as seen on the global BGP table.

One method growing in popularity to address the load balancing issue is to deploy BGP/LISP (Locator/Identifier Separation Protocol) gateways within an Internet exchange point to allow ingress traffic engineering across multiple links. This technique does not increase the number of routes seen on the global BGP table.

By design, routers running BGP accept advertised routes from other BGP routers by default. This allows for automatic and decentralized routing of traffic across the Internet, but it also leaves the Internet potentially vulnerable to accidental or malicious disruption, known as BGP hijacking. Due to the extent to which BGP is embedded in the core systems of the Internet, and the number of different networks operated by many different organizations which collectively make up the Internet, correcting this vulnerability (such as by introducing the use of cryptographic keys to verify the identity of BGP routers) is a technically and economically challenging problem.[34]

An extension to BGP is the use of multipathing – this typically requires identical MED, weight, origin, and AS-path although some implementations provide the ability to relax the AS-path checking to only expect an equal path length rather than the actual AS numbers in the path being expected to match too. This can then be extended further with features like Cisco's dmzlink-bw which enables a ratio of traffic sharing based on bandwidth values configured on individual links.

Multiprotocol Extensions for BGP (MBGP), sometimes referred to as Multiprotocol BGP or Multicast BGP and defined in IETF RFC 4760, is an extension to (BGP) that allows different types of addresses (known as address families) to be distributed in parallel. Whereas standard BGP supports only IPv4 unicast addresses, Multiprotocol BGP supports IPv4 and IPv6 addresses and it supports unicast and multicast variants of each. Multiprotocol BGP allows information about the topology of IP multicast-capable routers to be exchanged separately from the topology of normal IPv4 unicast routers. Thus, it allows a multicast routing topology different from the unicast routing topology. Although MBGP enables the exchange of inter-domain multicast routing information, other protocols such as the Protocol Independent Multicast family are needed to build trees and forward multicast traffic.

Multiprotocol BGP is also widely deployed in case of MPLS L3 VPN, to exchange VPN labels learned for the routes from the customer sites over the MPLS network, in order to distinguish between different customer sites when the traffic from the other customer sites comes to the Provider Edge router (PE router) for routing.

BGP4 is standard for Internet routing and required of most Internet service providers (ISPs) to establish routing between one another. Very large private IP networks use BGP internally. An example is the joining of a number of large Open Shortest Path First (OSPF) networks, when OSPF by itself does not scale to the size required. Another reason to use BGP is multihoming a network for better redundancy, either to multiple access points of a single ISP or to multiple ISPs.

Routers, especially small ones intended for Small Office/Home Office (SOHO) use, may not include BGP software. Some SOHO routers simply are not capable of running BGP / using BGP routing tables of any size. Other commercial routers may need a specific software executable image that contains BGP, or a license that enables it. Open source packages that run BGP include GNU Zebra, Quagga, OpenBGPD, BIRD, XORP, and Vyatta. Devices marketed as Layer 3 switches are less likely to support BGP than devices marketed as routers, but high-end Layer 3 Switches usually can run BGP.

Products marketed as switches may or may not have a size limitation on BGP tables, such as 20,000 routes, far smaller than a full Internet table plus internal routes. These devices, however, may be perfectly reasonable and useful when used for BGP routing of some smaller part of the network, such as a confederation-AS representing one of several smaller enterprises that are linked, by a BGP backbone of backbones, or a small enterprise that announces routes to an ISP but only accepts a default route and perhaps a small number of aggregated routes.

A BGP router used only for a network with a single point of entry to the Internet may have a much smaller routing table size (and hence RAM and CPU requirement) than a multihomed network. Even simple multihoming can have modest routing table size. See RFC 4098 for vendor-independent performance parameters for single BGP router convergence in the control plane. The actual amount of memory required in a BGP router depends on the amount of BGP information exchanged with other BGP speakers and the way in which the particular router stores BGP information. The router may have to keep more than one copy of a route, so it can manage different policies for route advertising and acceptance to a specific neighboring AS. The term view is often used for these different policy relationships on a running router.

If one router implementation takes more memory per route than another implementation, this may be a legitimate design choice, trading processing speed against memory. A full IPv4 BGP table as of August 2015 is in excess of 590,000 prefixes.[25] Large ISPs may add another 50% for internal and customer routes. Again depending on implementation, separate tables may be kept for each view of a different peer AS.