A short Introduction to BGP

and Cisco configuration hints

Andrea Pasquinucci

UCCI.IT

0. Abstract

In this short document, we introduce the basic concepts of BGPv4 and give some indications of how to implement a simple configuration on Cisco IOS routers.

Table of Contents

1. What is BGP

The Border Gateway Protocol (BGP) version 4, is the protocol adopted by what is today called internet to establish global routes so that IP packets know which route they should take to reach the destination. Thus BGP:

- is a *routing* protocol
- concerns only the *destination* of a packet, that is only the destination address
- is a local protocol, in the sense that each router keeps its own tables and knows only which is the *best next-hop* towards a final destination but not the full path to the destination
- is a global protocol, in the sense that its main use is to distribute all routes to all IP destinations on earth, even if it can be used just within an organization if one wants.

Thus all Internet Providers (ISP) and Carriers adopt BGPv4 to exchange IP traffic.

2. Principles of BGP

The basic principles of BGP are rather simple, even if their implementation is not at all trivial.

2.1 Autonomous Systems (AS)

An Autonomous System (AS) is an organization which is connected to internet using at least two different providers. In this case the organization is assigned a number called AS number (or ASN) which identifies it. Moreover the Organization is assigned a set of IP numbers which it can use as public internet numbers, that is the Organization can assign these numbers to its own public computers and all world internet traffic to these IP numbers will be routed to its own computers.

The numbers are usually divided in classes, adopting a notation called CIDR (Classless InterDomain Routing). For IPv4 IP numbers are 32 bits usually denoted in 4-dotted notation like 1.2.3.4/8 where the first 4 numbers run between 0 and 255 (i.e. 8 bits per number), and the last number after the slash denotes how many bits of the first 4 numbers denote the Network part and is between 0 and 32 (in this example: 1 is the Network part and 2.3.4 the Host part). It is important to note that IP numbers are divided in their Network and Host part, all numbers which share the same network part belong to the

same Organization and then to the same AS. In this way for Internet routing it is enough to specify the routes to the network part of the IP addresses, whereas the Host part is useful only within the internal network of the Organization.

2.2 Exterior and Interior BGP

The basic function of BGP is to exchange *BGP Tables* with other BGP speakers, usually routers but also route-servers. We will see in the next section some details of what is a BGP Table, the idea is that in a BGP Table there are AS Numbers with associated IP networks and indications of how to reach them. In other words, suppose that two routers belonging to different AS start a BGP session, each one will send to the other its own BGP Tables in which a router tells the other which network can be reached through itself. In other words, router A sends to router B a table of IP networks that router B can reach through router A, or even more clearly, router A says to router B "I am connected (directly or indirectly) to the following IP Networks, so you can send packets for these Networks to me".

This is usually called Exterior BGP, that is a BGP session between two peers belonging to different AS. In some cases, it is also necessary to establish BGP sessions between routers within the same AS, in this case it is an Interior BGP session. The typical case is when an organization has different connections to other AS in different locations and through different routers. Thus router A in a location and B in another location are both BGP routers with external peers. To balance the internal/external traffic with the external peers, the two routers must exchange their BGP informations and they can do this using I-BGP. Moreover, if the internal network is reasonably complicated, there will be also an internal routing protocol, like EIGRP or OSPF, and again the BGP routers will need to exchange routes with the internal routing protocol both to inform them of how to reach the outside world and, in case, to learn which network there are inside which should be announced to the outside world.

We will not consider I-BGP or internal routing protocols in the rest of this short introduction.

2.3 BGP Tables

The core of BGP are the BGP Tables. We will not enter in too many details of what exactly is a BGP Table, but instead we will give an *ideal representation* of the kind of informations that are exchanged. In other words, we will use a schematical description which will help our understanding, even if it will depart from the real technical description of BGP.

The information that a router sends to its peers are in broad summary of 3 kinds:

1. AS Numbers
2. IP networks
3. parameters (attributes) mainly related to the peering between the two routers

We will summarize these informations as in the following table:

```
AS-N3 AS-N2 AS-N1 {
  1.2.0.0/16
  7.8.9.0/24
} parameter1=value1; parameter2=value2 ...
```

Table 1

First comes a list of AS Numbers, it would seems strange that there are more than one AS Number, but we will see the reason for this. At the moment it is important to know that the first (rightmost) ASN is the one which owns the IP network listed between the curly brackets. Moreover, each router cannot delete any ASN from the list,[1] but instead usually must *prepend* (to the left) at least one ASN to the list. Actually this list of ASN is an attribute (or parameter) called AS_PATH, like the ones listed at the end, but for its importance we list it first. The list of ASN is actually either an ordered list AS_SEQUENCE or a mathematical set called an AS_SET.

As already said, it follows the list of IP networks owned, or announced, by the first (rightmost) ASN. At the end there is a list of parameters (attributes) most of which are *local* to the two routers exchanging the informations. An incomplete list of parameters (attributes) is the following:

- BGP version (usually 4, previous version do not support CIDR)
- IP address of next-hop, that is the IP to which the receiving router should send all packets for the listed Networks
- metric or MED (Multi Exit Discriminator)
- community
- table version number

2.4 Connection between two BGP speakers

The BGP protocols uses a tcp connection on port 179. Prior to the starting of the BGP peering there must exist (for example configured by hand) a route which allows the IP traffic to flow between the two peers. Moreover if the next-hop announced by a BGP peer is different from itself, there must exist routes also to the next-hop IP address.

1 Except for aggregation of AS_PATH

Once the tcp connection is established, the BGP peers exchange first some basic informations about themselves and then **all** their BGP tables in use (i.e. the best paths), both the ones internal and the ones received from other BGP peers (unless there are filters to prevent this). Each peer has a unique table version number (for all peers) which is updated every time something changes in its BGP tables. When this happens, the BGP speaker sends to all its established peers only the updates and the new version number. Keepalive packets are exchanged between peers and notification packets are sent in response to error or in special situations.

Notice that the default behaviour is that a BGP speaker sends to a peer all routing informations that it knows *and uses*, except for the those received from that peer. This means that it announces all networks of all its other peers to each peer! In other words to a peer it says "you can send me traffic for all my other peers and I will forward it to them". Thus by default every BGP peer becomes a **transit AS** for all its peers. This is the correct configuration for a internet carrier, but surely not for a small ISP or a final customer, who really only wants to be able to access internet through two or more ISP and not to become a transit path for a potentially big part of internet traffic. To prevent this, filter must be applied to the received and sent announcements from one peer to another. Filters are also used to balance, as far as possible, the traffic, or to influence the *choice of best routes,* as we will see in the practical examples.

The fact that a BGP speaker sends to a peer only routing informations that it knows and uses, means that the routing information is always local, hop-by-hop. Each BGP peer does not have a full map of the path to get from one point to another of internet, the only global information known is the AS_PATH. The graph of AS_PATH is crucial in what it is used to prevent loops based on the topology so constructed.

3. BGP in action

In this section we will try to follow what happens when a BGP section is established. There are two main phases:

1. exchange of BGP tables
2. creation of local routes

3.1 Exchange of BGP tables

At the beginning of a BGP connections, the two peers exchange some informations, like BGP version number (usually 4), an IP number chosen by each router as its own identifier (usually the first or highest IP number of all its own interfaces), the ASN of the peer. After that, each peer sends to the other all tables. This can happen as follows.

Consider first Networks that are local (internal) to the BGP speaker that is announcing them. In this case the original table can be as follow:

```
{
  3.4.5.0/24
  7.8.9.0/24
} next-hop=3.4.5.1; ...
```

Table 2

Here we assume that the router sending the information has IP 3.4.5.1. Notice that the router is telling its peer to send all traffic for 3.4.5.0/24 and 7.8.9.0/24 to itself. Notice also that there is no ASN at the beginning of the table! This is correct, local routes are identified by a **null** ASN.

Now, when the BGP speaker sends this table to its peer it must prepend its own ASN just before dispatching it. In this way the local table has a null ASN but the table that the peer receives has only the ASN of the sender. Assuming that AS-100 is the sender and AS-200 is the receiver, the table received by AS-200 is

```
AS-100 {
  3.4.5.0/24
  7.8.9.0/24
} next-hop=3.4.5.1; ...
```

Table 3

Now assume that AS-100 has in memory a table of AS-300, which is another peer:

```
AS-300{
  13.44.56.0/24
  9.4.3.0/24
} next-hop=9.4.3.1; ...
```

Table 4

AS-100 will proceed as before, sending this table to AS-200 modifying this time the parameters and prepending its own ASN just before dispatching it. AS-200 will receive

```
AS-100 AS-300{
  13.44.56.0/24
  9.4.3.0/24
} next-hop=3.4.5.1; ...
```

Table 5

Table 3 and Table 5 are received by AS-200, they both specify the next-hop 3.4.5.1 which is the IP number of the AS-100 router peer of AS-200. Moreover from Table 5 we can read that we can reach the networks of AS-300 passing through AS-100. If AS-200 has no better way to reach AS-300, it will send all traffic for these destination to AS-100. Notice that the AS-PATH gives an idea of *how far* a destination is in terms of AS hops, which can be quite different from networks hops and even more different in kilometers or channel capacity!

A BGP speaker which does only what mentioned up to now is usually called a *BGP route server*, it just receives and redistributes BGP routing informations. Usually instead a router translates these informations in routes used by IP packets traveling through its interfaces.

3.1 Creation of local routes

Having all tables is obviously not enough if we want to use them to route the IP traffic. To translate BGP tables in IP routes, a complex algorithm must choose the *best route to a destination* avoiding loops etc. (remember that it is normal that the same network is announced by more than one peer, so we have to choose which one to use). The selection criteria are as follows:[2]

- If the indicated Next-hop in unreachable, discard this route
- among two different routes to the same network prefer the route with highest **weight**, which is a preference value local to each single router

2 This is taken from the Cisco implementation of BGPv4.

- among two different routes to the same network prefer the route with highest **local-preference**, which is a preference value local to all routers in the same AS and connected by I-BGP
- among two different routes to the same network prefer the route **originated by the peer** which has announced it
- among two different routes to the same network prefer the route with the **shortest AS-PATH**
- among two different routes to the same network prefer the route with the **lowest origin code** (IGP < EGP < INCOMPLETE)
- among two different routes to the same network prefer the route with the **lowest MED**, which is a parameter which is zero by default and distributed only to peer AS but not redistributed by them
- among two different routes to the same network prefer the route with **External path** over Internal path
- among two different routes to the same network prefer the route with the lowest IP address value for the **BGP router ID**.

This long list of choices is not even complete, but gives an idea of the process of selecting the best route to a destination in a reasonable fair way. Of course in practice often one wants to modify the default choice due to many possible reasons, the most common being that the default selection does not consider the type of traffic which is carried and then it ends that one link is full and the other is empty. To modify the default selection the most common parameters used are the local-preference and the AS-PATH. Giving a higher local-preference (usually the default is 100) to a route learned from a peer, implies that that route will be chosen, inserted in the routing table and announced to the other peers. To influence the incoming traffic one can prepend to the table announced to a peer more time its own ASN. In this way that peer and all the ones behind it will have an AS-PATH longer than the one offered by another peer, and the traffic towards us would flow (barring local-preference choices) through the other peer. As an example, suppose that AS-100 is receiving too much traffic from AS-200. This means that AS-100 customers are *closer* to AS-200 than to AS-300. To try to balance the traffic we can prepend AS-100 to the AS-100 announcements to AS-200. Thus AS-200 will receive the following table from AS-100

```
AS-100 AS-100 AS-100 {
  3.4.5.0/24
  7.8.9.0/24
} next-hop=3.4.5.1; ...

AS-100 AS-100 AS-100 AS-300{
  13.44.56.0/24
  9.4.3.0/24
} next-hop=3.4.5.1; ...
```

Table 6

whereas AS-300 will receive

```
AS-100 {
  3.4.5.0/24
  7.8.9.0/24
} next-hop=3.4.5.1; ...

AS-100 AS-200{
  99.24.1.0/24
} next-hop=3.4.5.1; ...
```

Table 7

Now suppose that AS-400 has peering with both AS-200 and AS-300. AS-400 will receive an announce of AS-100 from AS-300 of length 2 (AS-300 AS-100) and from AS-200 of length 4 (AS-200 AS-100 AS-100 AS-100) and in absence of local parameters it will chose to send the traffic to AS-300 whereas if the paths would have been of the same length it would have chosen AS-200 due to some other parameter later on in the selection process.

Many more complicated operations can be done by modifying the default BGP parameters.

4. What is missing

In this brief introduction to BGPv4 we have omitted many more advanced subjects. For completeness we list here just a few:

- **Groups:** peers can be grouped in groups for easier administration
- **Communities:** this is an attribute which allows to group classes of destinations together and apply routing decisions to all of them; it can be distributed to peers
- **Confederations:** this is a way of dividing an internal I-BGP area in many sub-AS for easier internal management; externally there will be always a single AS
- **Route reflectors:** this is another way of distributing routes in a large internal I-BGP
- **Route flap dampening:** this is related to a very serious in-stability problem of BGP; there are situations in which a router keeps changing idea on what is a best route, but forwarding to its peers the new best route makes the peers changing their selections which implies that the first router changes back its own etc. etc.; this is called *route flapping* and can lead to disasters; dampening means that if a route flaps too much, it is withdrawn for a period of time hoping that in the meantime the instability is solved.

Other aspects will be discussed in the practical examples in the next sections.

5. Configuring Cisco routers

In this short introduction we will consider a very simple case, an Organization with its own ASN, in this case AS-100, connected to two Internet Providers, AS-200 and AS-300. AS-100 is not a transit AS, thus it will not allow traffic from/to AS-200 to flow to/from AS-300 through itself, but there will be only traffic to/from AS-100 from/to AS-200 or to/fromAS-100 from/to AS-300.

5.1 Basic configuration

We assume that AS-100 has IP numbers 1.2.0.0/20 and that the router has the following interfaces:

- Interface-1 towards AS-200: 7.8.1.2
- Interface-2 towards AS-300: 4.9.3.6
- Loopback interface: 1.2.15.254/32
- Interface-3 internal LAN: 1.2.0.1/21

The starting configuration is given in Table 8.

```
ip route 1.2.0.0 255.255.240.0 Null0 200
ip route 5.16.7.0 255.255.255.0 4.9.3.5
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 1.2.0.0 mask 255.255.240.0
 !
 neighbor 7.8.1.1 remote-as 200
 neighbor 7.8.1.1 description First ISP
 neighbor 7.8.1.1 ebgp-multihop 4
 neighbor 7.8.1.1 update-source Loopback0
 neighbor 7.8.1.1 soft-reconfiguration inbound
 neighbor 7.8.1.1 version 4
 neighbor 7.8.1.1 maximum-prefix 200000
 !
 neighbor 5.16.7.9 remote-as 300
 neighbor 5.16.7.9 description Second ISP
 neighbor 5.16.7.9 ebgp-multihop 4
 neighbor 5.16.7.9 update-source Loopback0
 neighbor 5.16.7.9 soft-reconfiguration inbound
 neighbor 5.16.7.9 version 4
 neighbor 5.16.7.9 maximum-prefix 200000
```

Table 8

We now comment this configuration. First of all, there are at least 3 ways of announcing a network to the peers. The first is to redistribute it from an internal routing protocol like OSPF, the second to redistribute from the routes inserted by hand in the router ("ip route....") and the third is to configure directly BGP with the announcements that should be done. This last is the preferred way in simple situations. But to announce a route, BGP requires that it exists in the routing table of the router, for this reason we add a default backup (weight 200) route to Null0 for 1.2.0.0/20. Notice then that one of the peers is not directly connected, the IP 5.16.7.9 does not belong to any IP class on the interfaces of the router, thus we need to put an explicit route to reach it.

Then "router bgp 100" declare that we have ASN 100 and starts the bgp routing process. First we declare "no synchronization" otherwise bgp will not start until it receives some informations from an Internal Routing Protocol, so this line is always needed if you do not redistribute all routes from OSPF or EIGRP or similar. We then require to log on the router syslog all changes of bgp peers. Finally we announce our network, that is the one of AS-100. If there are more networks to announce the command "network ..." can be repeated.

Now it starts the configuration of each peer independently. The command is of the form "neighbor IP-addr-peer command". It is important to start with "remote-as NUM" which declares which is the expected and requested ASN of the peer. If the peer claims to have a different ASN the peering will fail.

Then for the second ISP we put "ebgp-multihop 4" (4 can be raised to 10 or more if needed), this declares that this peer is not directly connected and is compulsory in this case because by default BGP expects its peers to be directly connected. The "update-source Loopback0" tells BGP to send to the peers the IP address of the Loopback0 interface as next-hop and to use it as source address for all IP packets of the tcp connection of BGP. The peers must establish a peering with 1.2.15.254 and not 7.8.1.2 or 4.9.3.6. Using a Loopback interface is quite convenient since the IP address is not associated to a physical interface that can be stopped for some reason, thus the peering is independent of the interfaces through which it is done and it is then much more reliable (but the peer has to add an explicit route to this address as we did for AS-300). On the other hand this implies that also the first ISP has a not directly connected peering, since for the AS-200 the IP 1.2.15.254 does not appear in any class of IP on its router. Thus also for AS-200 we need the ebgp-multihop parameter. If "update-source" is not used, BGP will use the IP number of the interface from which the BGP session is established, 7.8.1.2 for AS-200 and 4.9.3.6 for AS-300.

Thus the local table of AS-100 is

```
{
  1.2.0.0/20
} next-hop=1.2.15.154;version=4;...
```

Table 9

The "soft-reconfiguration" parameter allows to update the BGP tables without canceling all of them from memory. This guarantees smooth operation but requires quite some memory. If it is not used, it could happen that the tables are canceled and routing stops for a moment while the routers updates its tables. It is strongly suggested to adopt this parameter and add more RAM in case.

The "version 4" parameter requires that peers speak only BGPv4, making a faster startup, and "maximum-prefix" puts a higher limit to the number of networks (prefixes) that the router will accept from that neighbor, this is to prevent that the memory gets filled up by mistake. As at the time of writing the number of prefixes of a full internet table is about 140000.

After having done this configuration, we can save it and look at the establishment of the BGP connections. Using the command

```
show ip bgp neighbor summary
```

we can see the status of the connections. If at the right hand it says "IDLE" it means that no tcp

connection has been established. If it says “ACTIVE” it means that the tcp connection has been established, but BGP has not fully started yet. This could be due to the time needed to agree on the parameters between the two routers or because the two routers do not agree on some parameters and then the BGP session is not accepted. If after some time the connection remains in the “ACTIVE” state it means that there is a configuration problem.

5.2 Filtering by prefix-list

The configuration in Table 8 has a major problem, it allows to send all announcements of AS-200 to AS-300 and vice-versa, so that it is a Transit AS configuration. To prevent that we need to introduce some filters. Filtering is not trivial and a bit confusing, so we will limit ourselves to two filtering procedures: route-maps and prefix-list.

Prefix-list are simpler and we start from them. They allow to filter on prefixes, that is networks (IP numbers). There can be a list for incoming announcements and one for outgoing announcements for each peer. A list can be used for more than one peer in case. First we create simple INGRESS and EGRESS prefix-list:[3]

```
ip prefix-list INGRESS seq 1 deny 0.0.0.0/0
ip prefix-list INGRESS seq 5 deny 0.0.0.0/8 le 32
ip prefix-list INGRESS seq 10 deny 10.0.0.0/8 le 32
ip prefix-list INGRESS seq 15 deny 127.0.0.0/8 le 32
ip prefix-list INGRESS seq 20 deny 172.16.0.0/12 le 32
ip prefix-list INGRESS seq 25 deny 169.254.0.0/16 le 32
ip prefix-list INGRESS seq 30 deny 192.0.2.0/24 le 32
ip prefix-list INGRESS seq 35 deny 192.168.0.0/16 le 32
ip prefix-list INGRESS seq 40 deny 198.18.0.0/15 le 32
ip prefix-list INGRESS seq 45 deny 224.0.0.0/3 le 32
ip prefix-list INGRESS seq 100 deny 1.2.0.0/20 le 32
ip prefix-list INGRESS seq 200 permit 0.0.0.0/0 ge 6 le 27
ip prefix-list INGRESS seq 300 deny 0.0.0.0/0 le 32
!
ip prefix-list EGRESS seq 5 permit 1.2.0.0/20
ip prefix-list EGRESS seq 300 deny 0.0.0.0/0 le 32
```

Table 10

The prefix-list are simple in concept. One has to choose a name for each list and each entry has a sequence number which specifies the order in which the list is checked. This makes it easy to add new

3 INGRESS and EGRESS are the names we have chosen for these prefix-lists.

entries where is needed. Deny and permit specify what to do if the entry matches, if the entry is permitted then the prefix (network) will be inserted in the local table if in input, or sent to the peer if in output. It follows the network to match in the notation x.y.z.w/n. If the line does not have other parameters, than the match must be exact, otherwise one can specify sub or super matches. This is explained easier with an example. The following

```
ip prefix-list INGRESS seq 10 deny 10.0.0.0/8
```

matches exactly 10.0.0.0/8, but what if we receive 10.0.0.0/9 and 10.128.0.0/9 ? Put together they are the same announcement. The following

```
ip prefix-list INGRESS seq 10 deny 10.0.0.0/8 le 32
```

will match 10.0.0.0/8 but also all 10.x.y.z/n with any n less-than-or-equal-to 32 and greater than 8 (the 8 comes from the mask of 10.0.0.0/8).

The two prefix-lists of Table 10 have the following meaning. The INGRESS list filters out private addresses that we should not receive from internet, and AS-100 addresses that nobody else should announce. Moreover it accepts all announcements which are from /6 to /27, denying both announcements which are too large and cover too many addresses, and those which are too small and will result in too big routing tables. These last are usually artifacts of redistributing routes from internal routing protocols without doing some summarization or aggregation of addresses. Usually it is safe to deny so small addresses because there is the announcement of a larger network which contains them.

The EGRESS list allows to send to the peers only the networks announced by AS-100, preventing it to become a transit AS.

To insert the prefix-list in the BGP session, we add the relative instructions to those of Table 8 which becomes:

```
ip route 1.2.0.0 255.255.240.0 Null0 200
ip route 5.16.7.0 255.255.255.0 4.9.3.5
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 1.2.0.0 mask 255.255.240.0
 !
 neighbor 7.8.1.1 remote-as 200
 neighbor 7.8.1.1 description First ISP
 neighbor 7.8.1.1 ebgp-multihop 4
 neighbor 7.8.1.1 update-source Loopback0
 neighbor 7.8.1.1 soft-reconfiguration inbound
 neighbor 7.8.1.1 version 4
 neighbor 7.8.1.1 maximum-prefix 200000
 neighbor 7.8.1.1 prefix-list INGRESS in
 neighbor 7.8.1.1 prefix-list EGRESS out
 !
 neighbor 5.16.7.9 remote-as 300
 neighbor 5.16.7.9 description Second ISP
 neighbor 5.16.7.9 ebgp-multihop 4
 neighbor 5.16.7.9 update-source Loopback0
 neighbor 5.16.7.9 soft-reconfiguration inbound
 neighbor 5.16.7.9 version 4
 neighbor 5.16.7.9 maximum-prefix 200000
 neighbor 5.16.7.9 prefix-list INGRESS in
 neighbor 5.16.7.9 prefix-list EGRESS out
```

Table 11

To push the new configuration to the peers we can use the commands

```
clear ip bgp 7.8.1.1 soft in
clear ip bgp 7.8.1.1 soft out
clear ip bgp 5.16.7.9 soft in
clear ip bgp 5.16.7.9 soft out
```

Table 12

Obviously each command can be given independently. Sometimes it is not enough to have a soft update of the BGP session but it is required to restart the session from zero. To restart a session with a peer just give the same command as in Table 12 but without the “soft in/out” parameter. Notice that in

this case all routes announced by the peer will be withdrawn and traffic with that peer will stop for the time it takes to restart the session. Notice that sometimes if the prefix-list are modified, it is needed to remove them form the BGP configuration and reload them again to make the modifications active.

Finally to see the contents of the BGP tables the following commands can be used

```
show ip bgp [prefix-num]
show ip bgp regexp [ASN regexp]
```

the first shows the routes in the BGP tables for the network indicated, the second shows all routes for the ASN indicated, as in “show ip bgp regexp 200”. In this second case it can be used a ASN regular expression which will be discussed in the next section.

5.3 Filtering by route-maps

Filtering by route-map is more complex but allows to filter and modify also AS-PATHs. The idea is to create first one or more filters of paths, attributes or prefixes (IP networks), then to apply to the selected routes or prefixes some actions. We will describe it with some examples. First of all we consider incoming tables and for those received from AS-200 we give a higher preference:

```
route-map as200-in permit 10
 set local-preference 150
```

Table 13

This route-map is called “as200-in”, its action is to “permit” the prefix that match the rules (the opposite would be a “deny”). The “10” at the end of the definition of the route-map is a sequence number, the route-map can be made up of more statements which are checked in order of sequence number, as we will see. In this case there is only one sequence. There is no “match” statement, so all prefixes and AS-PATHs match this rule, but there is a “set” statement which says what to do. To all prefix received from AS-200 it is given local-preference 150, which is higher than the default 100 and so these prefix will be preferred and chosen for the local routes.

Now for the outgoing route-map towards AS-200:

```
ip as-path access-list 11 permit ^$
!
route-map as200-out permit 10
 match as-path 11
```

Table 14

Opposite to the previous case, here there is a filter ("match") but not an action ("set"), this means that all AS-PATH which match the filter will be permitted, the others denied. The filter is on AS-PATHs and is given by the "ip as-path access-list 11" (11 is the number which identifies the "as-path access-list"). The access-list says to permit the AS-PATH which match the regular expression ^$, what does that mean? The answer is: locally generated prefixes, that is *null* AS-PATH as we have already seen. Anyway it is convenient to pause to explain the basic points of the AS regexp syntax. An AS regexp is a list of AS numbers to match. There are various operators that can be used in the matching:

Character/Symbol	Special Meaning
asterisk *	Matches 0 or more sequences of the pattern.
period .	Matches any single character, including white space.
plus sign +	Matches 1 or more sequences of the pattern.
question mark ?	Matches 0 or 1 occurrences of the pattern.
brackets []	Designates a range of single-character patterns, [ab] matches a or b.
hyphen -	Separates the end points of a range, [0-9] matches any digit.
caret ^	Matches the beginning of the input string.
dollar sign $	Matches the end of the input string.
underscore _	Matches a comma (,), left brace ({), right brace (}), left parenthesis, right parenthesis, the beginning of the input string, the end of the input string, or a space.
parentheses ()	Designates a group of characters as the name of a confederation.

To match a special character, one should precede it by a backslash "\", moreover [^] indicates a negated range, that is it matches everything except what is in the indicated range. Thus ^$ means the empty or null AS-PATH. A simple example of regexp is `^(100_)*$` which matches: `^$`, `^100$`, `^100 100$`, `^100 100 100$` etc. that is all local tables and all tables of AS-100 with only 100 as ASN.

We now pass to more complicated route-maps for AS-300.

```
access-list 22 deny 11.22.33.0 0.0.0.255
access-list 22 permit any
!
ip as-path access-list 22 permit ^300$
!
route-map as300-in permit 10
 match ip address 22
 match as-path 22
 set local-preference 200

route-map as300-in permit 20
 set local-preference 100
```

Table 15

The route-map as300-in has two instances. In the first, sequence number 10, we do a double match both on prefixes and AS-PATH. For prefixes we allow all prefixes except for 11.22.33.0/24 as specified by the access-list 22, for AS-PATH we allow only AS-300. In practice what we do is to allow only the prefixes originated in AS-300 and not from other AS behind AS-300, with the exception of 11.22.33.0/24. To all these prefixes we give local-preference 200, that is higher than the one given before to those of AS-200. This makes sense. AS-300 is directly connected and thus it is meaningful to send direct traffic to it, but we want all other traffic to go through AS-200 included the traffic for 11.22.33.0/24. The second instance of the route-map gives preference 100 to all routes announced by AS-300, this is needed in case the connection with AS-200 would fall and we will need all traffic to go through AS-300. The outgoing route-map towards AS-300 is

```
ip as-path access-list 21 permit ^$
!
route-map as300-out permit 10
 match as-path 21
 set as-path prepend 100
```

Table 16

In this case we permit to send to AS-300 announcements which contain the null AS-PATH, i.e. the locally generated routes, and we prepend to our announcement an extra “100” so that we will get more incoming traffic from AS-200 than from AS-300 since the AS-PATH announced to others by AS-300 is longer.

The final configuration of BGP is given by:

```
ip route 1.2.0.0 255.255.240.0 Null0 200
ip route 5.16.7.0 255.255.255.0 4.9.3.5
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 1.2.0.0 mask 255.255.240.0
 !
 neighbor 7.8.1.1 remote-as 200
 neighbor 7.8.1.1 description First ISP
 neighbor 7.8.1.1 ebgp-multihop 4
 neighbor 7.8.1.1 update-source Loopback0
 neighbor 7.8.1.1 soft-reconfiguration inbound
 neighbor 7.8.1.1 version 4
 neighbor 7.8.1.1 maximum-prefix 200000
 neighbor 7.8.1.1 prefix-list INGRESS in
 neighbor 7.8.1.1 prefix-list EGRESS out
 neighbor 7.8.1.1 route-map as200-in in
 neighbor 7.8.1.1 route-map as200-out out
 !
 neighbor 5.16.7.9 remote-as 300
 neighbor 5.16.7.9 description Second ISP
 neighbor 5.16.7.9 ebgp-multihop 4
 neighbor 5.16.7.9 update-source Loopback0
 neighbor 5.16.7.9 soft-reconfiguration inbound
 neighbor 5.16.7.9 version 4
 neighbor 5.16.7.9 maximum-prefix 200000
 neighbor 5.16.7.9 prefix-list INGRESS in
 neighbor 5.16.7.9 prefix-list EGRESS out
 neighbor 5.16.7.9 route-map as300-in in
 neighbor 5.16.7.9 route-map as300-out out
```

Table 11

6. Some security issues

BGP, together with DNS, is one of the critical infrastructure elements of internet. Without BGP IP traffic will travel only locally. It is somehow a miracle that the infrastructure has shown to be so robust, since it has not been created with resilience and security as first objective. There are two main points of attacks towards BGP:

1. external attacks, like Denial of Service
2. internal attacks, where for example false informations are transferred using BGP to divert traffic.

The first kind of attacks are mitigated by the fact that the establishment of a BGP session requires that the configuration on both peers be specular. In other words, a BGP peer does not accept a new session if it does not have the correct and correspondent information in its own configuration. As a simple example, suppose that a router with AS-400 and IP number 33.67.8.9 tries to establish a BGP peering with our router AS-100. No peer with such data exists in the configuration of AS-100, and thus no BGP session is established. Actually, in practice it often happens that there are problems in the establishment of the peering due to different parameters in the two routers, so even if one wants it can be difficult to have a BGP session started at all.

If the establishment of a new BGP session is not possible without previous configuration and this prevents many possible attacks, BGP is open to Denial of Service and to spoofing attacks. Spoofing attacks are those in which someone inserts some tcp packets in the communication between two peers. It could even been a Man-in-the-Middle attack in which tcp packets between the two peers are changed by someone in the middle. This could lead to changes in the BGP tables, and thus in the routing of IP packets, or to Denial of Service, since the attacker could decide to tell one router to shut down BGP.[4] To defend against these kind of attacks it is possible to adopt MAC signature to all tcp packets [2]. In practice the peers share a secret key, and to each BGP tcp packet is added a MD5 MAC Signature done by putting together most of the tcp packet and the secret key and computing a MD5 hash of it which is appended to the packet. The receiver, before examining the packet, recomputes the MD5 MAC since it knows the secret key and verifies it with the one in the packet, if all is well the packet is accepted and passed to the BGP process. In this way spoofing and MitM attacks can be mitigated (but not completely avoided) even if MD5 MAC and fixed shared secret password are not very secure (to have better security the password should be changed regularly and MD5 traded for SHA1). In Cisco the implementation is quite simple, is enough to add `neighbor 7.8.1.1 password secret-password` to the configuration of a peer.

4 This attack, based on using the RST or SYN tcp flags, has made the headlines of all major newspapers in April 2004.

In any case, pure DoS remain a problem for BGP. If an attacker floods tcp port 179 of a peer, soon the BGP sessions will fail due to the missing keepalive between the peers. This is a generic DoS attack and there is little to do from the point of view of BGP. Of course the peers could protect tcp port 179 to reduce traffic towards them, but then a generic DoS on a router could stop BGP and all traffic.

The good point about BGP it is that is a local protocol, so that even if one peer fails, the whole network will experience in case some instabilities but it should soon adjust to the new configuration. In other words, local problems should not be distributed (if flapping and instabilities are correctly treated). Still BGP is in-band, that is uses the same channel as the traffic that it guides, so a problem in the traffic could lead to a problem in BGP which then can not lead the traffic anymore. This is not to say that the BGP infrastructure is safe due to its locality. A well aimed DoS attack to crucial BGP peers could crush internet.

The second kind of attacks are within the BGP protocol, in practice the distribution of false information to force routers to have wrong routes and thus to send IP packets not to the legitimate destinations. Unfortunately this happens both because of configuration errors and because of real attacks.[5] In ref. [3] there are statistics on illegal announcements which are found in BGP tables, from private IP networks (prefixes), to private AS-numbers and to networks announced by more than one AS ! So what happens when a prefix is announced by *two* different AS ? How can a router decide automatically which is the true AS which owns those IP numbers ? The only way to defend against fake announcements is to establish filters on the BGP tables received by the peers. We have already shown the simplest filters with the INGRESS and EGRESS prefix-list. Other filters should be adopted to limit the possibilities of receiving, and sending, wrong informations and a good help can be given by establishing a BGP session with the *Cymru Bogon Route Server Project* [4].

7. Bibliography

[1] Border Gateway Protocol 4 (BGP-4), RFC 1771

[2] Protection of BGP Sessions via the TCP MD5 Signature Option, RFC 2385

[3] *The Team Cymru BGP Data Page,* http://www.cymru.com/BGP/index.html

[4] *The Team Cymru Bogon Route Server Project,* http://www.cymru.com/BGP/bogon-rs.html

5 One should not forget the possibility of a router taken by an attacker who modifies the BGP announcements this router sends to its peers; unfortunately it is too often very easy to have access to a router.