

BGP Refresher



Network Engineering Workshop UC2021



The AfricaConnect3 project receives funding from the European Union under Grant Contracts DCI-PANAF/2019/411-583/584/585/586



Outline



- How we got here
- The evolution of BGP
- BGP messages and states
- BGP Attributes
- Dual Homed and Multihomed designs
- Why routers ignore BGP paths



How we got here

- The National Science Foundation Network (NSFNET) was an umbrella of projects sponsored by the National Science Foundation (NSF) from 1985 to 1995 to promote advanced research and education networking in the United States.
- NSFNET used Exterior Gateway Protocol version 2 (EGP-2) and Policy Based Routing.
 - EGP was viewed as a short term measure until better inter autonomous system protocols are defined and widely deployed by regional networks.

Border Gateway Protocol



- Also known as the “three napkins protocol”, the Border Gateway Protocol (BGP) was originally invented in 1989 by Kirk Lougheed (Cisco) and Yakov Rekhter (with IBM at that time), with the help of Lenn Bosak (with Cisco at that time).
- It was built on experience gained with the design and usage of EGP in the NSFNET backbone.



Border Gateway Protocol - TNP

B.G.P. Boundary Gateway Protocol

block length	2 bytes
version number	1 byte
block type	1 byte
holddown timer	2 bytes (minutes)

types:

- open - 1
- update - 2
- notification - 4
- keepalive - 8

version is currently 1

open:

- my AS # - 2 bytes
- link type - 1 byte
 - up - 1
 - down - 2
 - internal - 4 (not used in update database field)
 - H-link - 8
- auth type code - 1 byte
 - 0 - none
- authentication - variable

update:

- network # - 4 bytes
- first hop gateway - 4 bytes
- metric - 2 bytes
- cost of AS - 1 byte
- direction - 1 byte
- AS # - 2 bytes

report "route" times

notification:

- code - 2 bytes
- data - variable

- link type error in open - my view of current link type (1 byte)
- unknown auth type code - no data
- authentication failure (no data)
- update error - data in block is error

~~routing trap in update~~

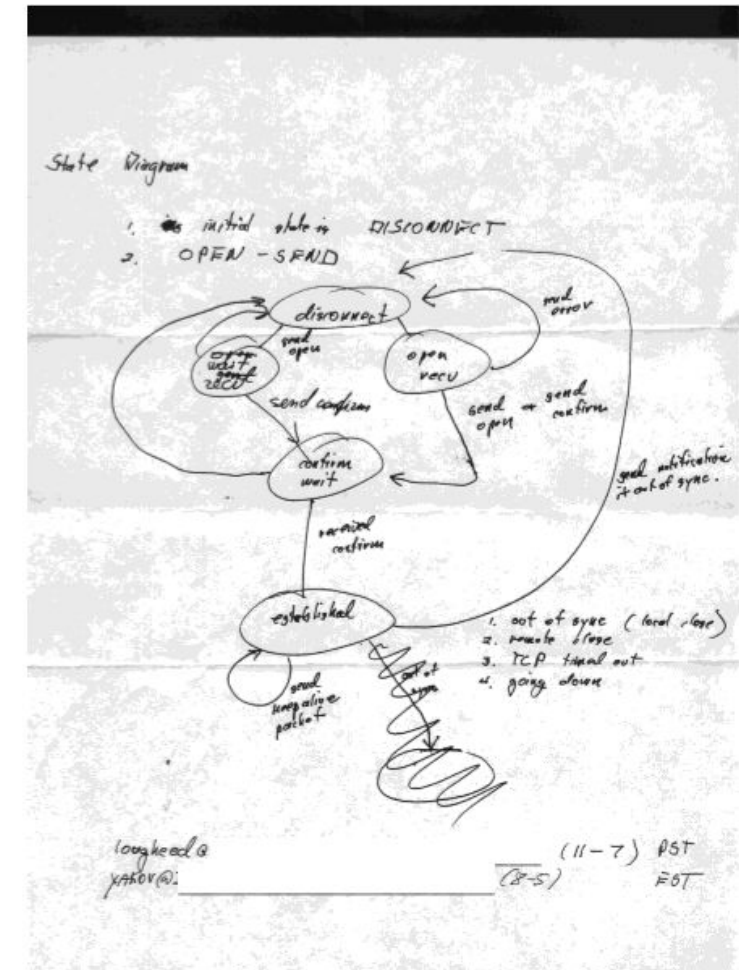
Two phase - receive & update

data is subroute (2 byte) followed by update block in question (1 network only)

subroutes -

- invalid network field
- invalid first hop gw
- invalid direction code
- invalid AS
- routing loop
- two-phase error

- connection out of sync - data is last block received (TCP close after packet sent)
- open confirmed
- invalid block type (data is 1 byte block type)
- invalid version number (data is 1 byte version)



What were the original design goals?

- To overcome limitations of EGP-2
 - Eliminate the restriction on inter-AS topology to be spanning tree
 - Eliminate problems caused by fragmentation of EGP-2 updates
- To support a few thousand classful IPv4 routes
- To replace EGP-2 in the NSFNET backbone

Evolution of BGP

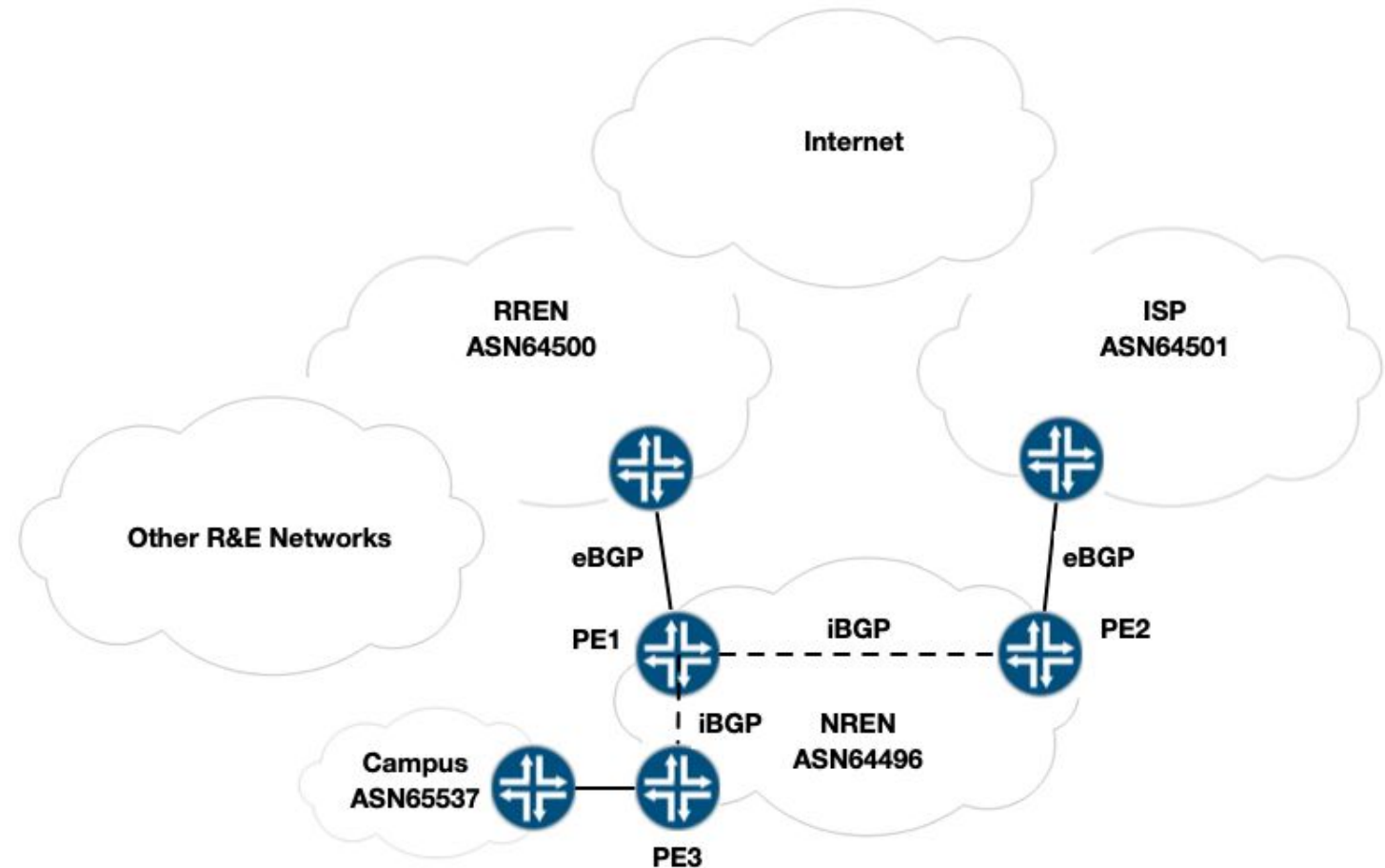
- BGP-1
 - Use TCP/179 unicast as a reliable transport protocol
 - Incremental updates instead of periodic
 - Have Autonomous Systems make up AS_PATHs
 - RFC1105 - Published in June 1989
- BGP-2
 - Introduced Path Attributes
 - Introduced the marker field
 - RFC1163 - Published in June 1990

- BGP-3
 - Optimise and simplify the exchange of information about previously reachable routes
 - Restrict a pair of speakers to a single BGP session
 - RFC1267 - Published October 1991
- BGP-4
 - Support CIDR
 - Introduced the LOCAL_PREF Attribute
 - RFC1771 - Published March 1995

- The first BGP specification was published in 1989, before IPv6 was created and only shortly after multicast was added to IPv4.
 - The original BGP-4 doesn't support IPv6, multicast or VPNs.
- In 1998, RFC2283 introduced the multiprotocol extensions.
 - These allow BGP to handle routing information for multiple address families
- In practice, an address family is associated with a specific network layer protocol, such as IPv4, IPv6, IPX or AppleTalk.
 - There are also Address Family Identifier (AFI) numbers for tunneling mechanisms such as VPNs and MPLS.
 - Subsequent Address Family Identifiers (SAFI) provide additional information for some address families.

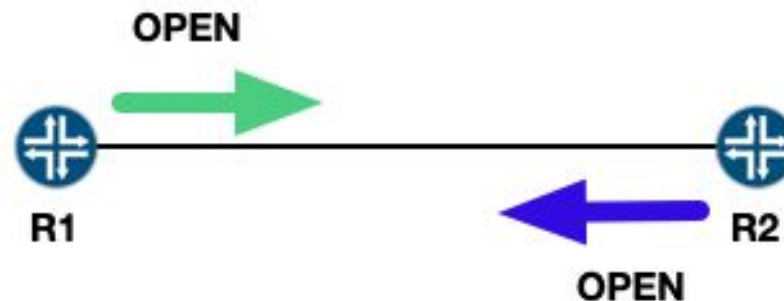
BGP Types

- External BGP (eBGP) is peering between routers in different ASes
 - Use physical interface IP addresses
- Internal BGP (iBGP) is peering between routers inside an AS.
 - Use loopback IP addresses



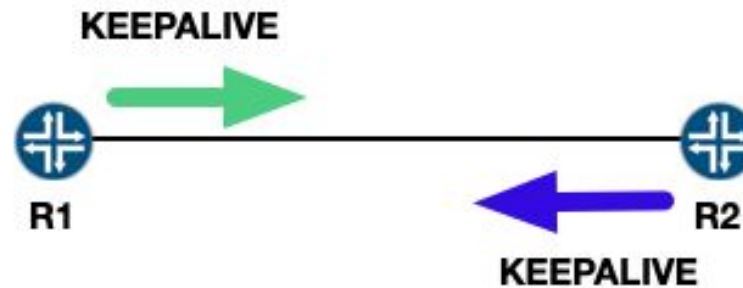
BGP Open Messages

- Used to exchange information required to establish a neighbour relationship
- It contains the BGP version, ASN of the speaker, hold time, router ID, optional parameters.



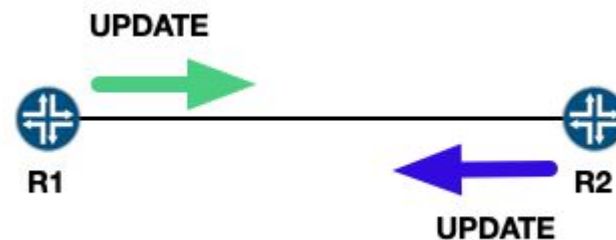
BGP Keepalive Message

- These messages are used to let a neighboring router know the other end is fine, but just didn't have any updates to send.
- They are sent $\frac{1}{3}$ of the negotiated holdtime.



BGP Update Messages

- The update message carries three types of information:
 - Network layer reachability information (NLRI). The NLRI field is simply a list of prefixes that are advertised as being reachable
 - A set of path attributes. This field holds information about those advertised prefixes.
 - A list of withdrawn routes. If previously advertised prefixes are no longer reachable, they're sent as withdrawn routes in an update.



BGP Notification Messages

- These are used to signal a BGP error and typically result into a reset of the BGP neighbourhood.
- The message contains
 - Error code
 - Sub-code
 - Data

BGP Route Refresh Messages (Optional)



- These messages are used to dynamically request a BGP router to resend Update messages
- Only sent if the two BGP speakers support the route refresh capability



- **Idle**

- This is the first state where BGP waits for a start event. A start event occurs when a new BGP neighbor is configured or when an established BGP peering is reset.
- The router initiates a TCP connection with its configured neighbour
- When successful, BGP moves to the Connect state. When it fails, it will remain in the Idle state.

- **Connect**

- BGP is waiting for the TCP three-way handshake to complete.
- If it is successful, BGP will send an OPEN message and continue to the OpenSent state.
- In case it fails, BGP continues to the Active state.
- If the ConnectRetry timer expires then the router will remain in this state.

- **Active**

- BGP will try another TCP three-way handshake to establish a connection with the remote BGP neighbor.
- If it is successful, it will move to the OpenSent state.
- If the ConnectRetry timer expires then BGP moves back to the Connect state.

- **OpenSent**

- BGP will be waiting for an Open message from the remote BGP neighbor.
- The open message will be checked for errors and if something is wrong (incorrect version numbers, wrong AS number, etc.) then BGP will respond with a Notification message and return to the Idle state.
- If everything is OK then BGP starts sending keepalive messages and resets its keepalive timer.

- ***OpenConfirm***
 - BGP waits for a keepalive message from the remote BGP neighbor.
 - When the local router receives the keepalive, it moves to the established state and the neighbor adjacency will be completed. When this occurs, the router reset the hold timer.
 - If local router receives a notification message from the remote BGP neighbor then it falls back to the Idle state.
- ***Established***
 - The BGP neighbor adjacency is complete and the BGP routers will send update packets to exchange routing information.
 - The hold timer will be reset every time a router receives a keepalive or update message.
 - In case the router receives a notification message, it will jump back to the Idle state.

- Unlike IGPs that work on the principle of finding the shortest path to a destination, BGP uses a set of attributes to determine the best path for each destination.
- ISPs usually implement policies by modifying route attributes and changing the way routers react to advertisements with certain route attributes.
- BGP attributes include:
 - WEIGHT
 - AS_PATH
 - MED
 - Local_Pref

Why Router Reject/Ignore Routes

- Paths for which the NEXT_HOP is inaccessible.
 - Ensure there is an IGP route to the NEXT_HOP that is associated with the path.
- Paths from an external BGP neighbor if the local autonomous system appears in the AS path.
- Paths that are denied by a routing policy implemented via access-lists, rout-filters, prefix-lists, AS_PATH-lists or community-lists.

Why Router Reject/Ignore Routes

- If paths are marked as not synchronised.
 - The BGP synchronisation rule states that if an AS provides transit service to another AS, BGP should not advertise a route until all of the routers within the transit AS have learned about the route via an IGP.
 - You can disable synchronisation if one of the following conditions is true
 - All the transit routers in your AS run BGP
 - Your AS does not pass traffic from one AS to another AS
 - You have an MPLS enabled backbone

Single/Dual Homed vs Multihoming

- Single homed
 - This means the customer has one connection to the ISP.
- Dual homed
 - This design adds redundancy. The customer uses two links to connect to a single ISP.
- Multihomed
 - Here the customer is connected to at least two different ISPs.

Configuration Example - IOS-XE

```
!  
router bgp 64496  
  bgp log-neighbor-changes  
  neighbor 2001:DB8:1001:0:100:104:1:1 remote-as 64496  
  neighbor 2001:DB8:1001:0:100:104:1:1 update-source Loopback0  
  neighbor 2001:DB8:1002::5 remote-as 65536  
  neighbor 100.104.1.1 remote-as 64496  
  neighbor 100.104.1.1 update-source Loopback0  
  neighbor 100.104.2.10 remote-as 65536  
!  
  address-family ipv4  
    network 100.104.128.0 mask 255.255.255.0  
    neighbor 100.104.1.1 activate  
    neighbor 100.104.1.1 next-hop-self  
    neighbor 100.104.2.10 activate  
  exit-address-family  
!  
  address-family ipv6  
    network 2001:DB8:1801::/48  
    neighbor 2001:DB8:1001:0:100:104:1:1 activate  
    neighbor 2001:DB8:1001:0:100:104:1:1 next-hop-self  
    neighbor 2001:DB8:1002::5 activate  
  exit-address-family  
!
```

Configuration Example - Junos

```
set protocols bgp group NRENA-iBGP-V4 type internal
set protocols bgp group NRENA-iBGP-V4 local-address 100.104.1.11
set protocols bgp group NRENA-iBGP-V4 export iBGPv4-OUT
set protocols bgp group NRENA-iBGP-V4 local-as 64496
set protocols bgp group NRENA-iBGP-V4 neighbor 100.104.1.1
set protocols bgp group NRENA-iBGP-V6 type internal
set protocols bgp group NRENA-iBGP-V6 local-address 2001:db8:1001::100:104:1:11
set protocols bgp group NRENA-iBGP-V6 export iBGPv6-OUT
set protocols bgp group NRENA-iBGP-V6 local-as 64496
set protocols bgp group NRENA-iBGP-V6 neighbor 2001:db8:1001::100:104:1:1
set protocols bgp group NRENA-eBGP-V4 export eBGPv4-OUT
set protocols bgp group NRENA-eBGP-V4 local-as 64496
set protocols bgp group NRENA-eBGP-V4 neighbor 203.0.113.33 peer-as 64500
set protocols bgp group NRENA-eBGP-V6 export eBGPv6-OUT
set protocols bgp group NRENA-eBGP-V6 local-as 64496
set protocols bgp group NRENA-eBGP-V6 neighbor 2001:db8:a002:: peer-as 64500
```

Configuration Example - Junos (cont..)

```
set policy-options policy-statement iBGPv4-OUT term 10 from protocol bgp
set policy-options policy-statement iBGPv4-OUT term 10 then next-hop self
set policy-options policy-statement iBGPv4-OUT term 10 then accept
set policy-options policy-statement iBGPv4-OUT term LAST then reject

set policy-options policy-statement iBGPv6-OUT term 10 from protocol bgp
set policy-options policy-statement iBGPv6-OUT term 10 then next-hop self
set policy-options policy-statement iBGPv6-OUT term 10 then accept
set policy-options policy-statement iBGPv6-OUT term LAST then reject
```

Thank you

Any questions?



@AC3_News



AC3-Community



africconnect3.net



The AfricaConnect3 project receives funding from the European Union under Grant Contracts DCI-PANAF/2019/411-583/584/585/586

