



# JANET

## IPv6 Hands-on Workshop

Module 6: **Interdomain Routing (BGP)**

UKERNA, Lancaster University  
and University of Southampton, 2006



# Interdomain routing

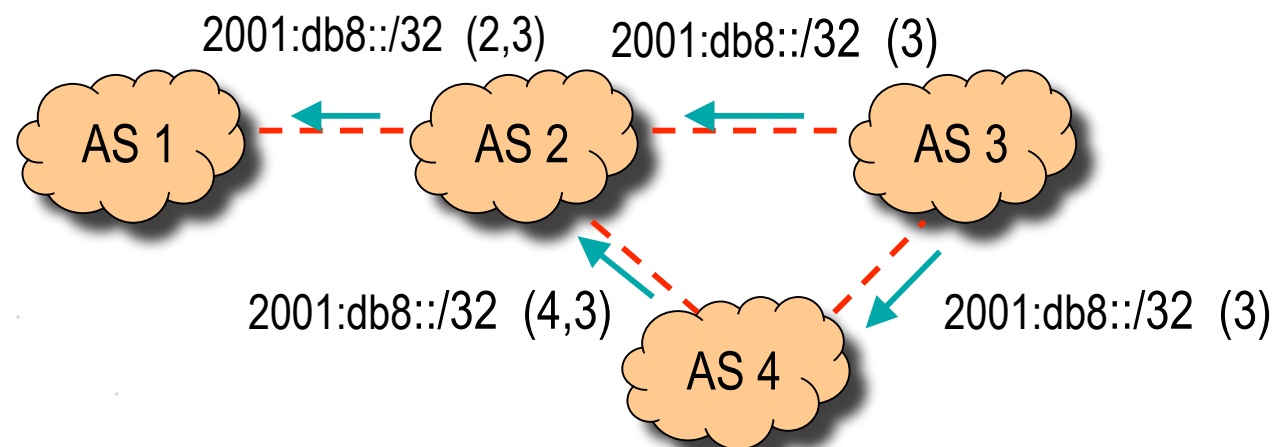
- While IPv6 has much greater address space, the principal method for interdomain routing remains the same
  - Both IPv4 and IPv6 use Border Gateway Protocol (BGP)
- For a site or network using BGP, you will need to know
  - How to configure BGP for IPv6
  - How to run IPv4 and IPv6 BGP together
  - What prefix filtering policies you may wish to apply
  - How to filter routes in BGP
- We look at these specifics in this theory session
  - You can then configure BGP in the lab

# BGP basic principles

- BGP is the routing protocol used between domains
  - A domain is a part of the Internet that is under the same technical administration
  - Inside each such domain one uses internal routing protocols like OSPF/IS-IS
  - BGP is used between e.g. JANET and GEANT, between JANET and regional networks, and also possibly between regional networks and sites. Each of these are separate management domains
- Domains are assigned AS (Autonomous System) numbers (ASNs)
- Domains set up BGP peerings between each other
- Domains can offer transit to others by passing on advertised routes that they received from one peer onto another
- Routers end up with a list of destination prefixes, one or more AS paths for each, and the next-hop router for each
- The general idea is to use the shortest AS path

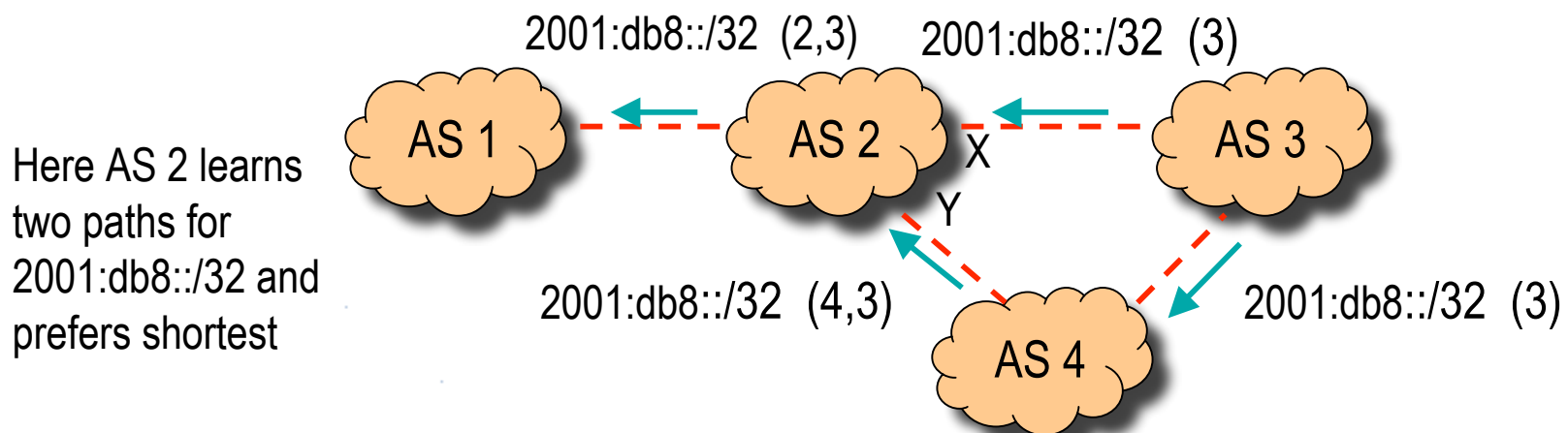
# BGP - example

- In this example there are four Autonomous Systems (ASes)
- Here, AS 3 is advertising 2001:db8::/32 to AS 2 and AS 4
  - AS 2 learns two paths for 2001:db8::/32 and prefers the shortest
    - One path via AS 4 to AS 3
    - One path directly from AS 3
  - AS 1 sees its path for 2001:db8::/32 via AS 2 to AS 3 (i.e. 2,3)



# Internal BGP – iBGP

- A domain with multiple exit routers may use internal BGP (iBGP) between the exit routers
  - The exit routers must agree on which has the best route for each of the external prefixes
  - Each exit router may then announce into OSPF/IS-IS which external prefixes it is preferred for (to inform through iBGP which next-hop will be used by the other, and you use the IGP to reach that next-hop, so IGP only manages internal addresses).
- Assume below that AS 2 has one exit router X facing AS 3 and another exit router Y facing AS 4
  - Both X and Y will have a route for 2001:db8::/32
  - Since X has shortest path to AS 3, we want X to be next-hop for 2001:db8::/32 internally
  - With an iBGP peering between X and Y, they can agree who has the best route



# BGP version 4

- BGP version 4 specified in RFC 1771
  - It only supports IPv4
- The IPv4 specific parts are
  - NLRI (Network Layer Reachability Information) in an UPDATE message contains an IPv4 prefix (the announced prefixes)
  - NEXT\_HOP path attribute in an UPDATE message is an IPv4 address
  - BGP Identifier in OPEN message and AGGREGATOR attribute
- In order to support other address families than just IPv4, these parts must be changed
- This model also supports only unicast
  - When doing multicast one often has different topologies for unicast and multicast, so there is also a need for learning multicast routes (used for RPF)
- So people wanted to extend the original BGP spec for several reasons

# Multiprotocol BGP – MBGP

- Multiprotocol Extensions for BGP-4, specified in RFC 2858
- Adds MP\_REACH\_NLRI and MP\_UNREACH\_NLRI
  - Unreachable status allows the quick withdrawal of routes
- These include AFI (Address Family Identifier) and SAFI (Subsequent Address Family Identifier) fields
- AFI tells whether it's IPv4 or IPv6
- SAFI can tell whether it's unicast, multicast (for RPF) etc
  - RFC 2858 defines SAFI 1 for unicast, 2 for multicast and 3 for both
    - The IETF is working on an update of this without SAFI 3. We are not aware of any implementations using 3.
    - Other values are proposed in other RFCs/drafts for MPLS, VPNs etc

# More MBGP features

- Next hop in MP\_REACH\_NLRI, IPv4 or IPv6 according to AFI value
  - For IPv6, next hop always includes global IPv6 address
  - Next hop includes link-local IPv6 address if and only if the peers are on same link
    - Often not the same link with iBGP and one can also do multi-hop BGP
- The identifier is still IPv4, assuming each router has at least one IPv4 address
  - If not, the identifier can be set manually to a 32-bit number
- Together, all these features allow BGP to be used for IPv6 and for multicast

# IOS BGP configuration

- In the past IOS BGP configuration was all done directly beneath “router bgp <as>”
  - “nlri” keyword was used to indicate multicast
- Now there are subsections for different families. So you may have “address-family ipv4 [unicast]” and “address-family ipv4 multicast”
- For IPv6 we follow the same pattern with “address-family ipv6 [unicast]” and “address-family ipv6 multicast”
- If you use peer groups, the group names for IPv4 and IPv6 peers must be different
- It’s possible to exchange IPv6 routing using IPv4 transport, but we suggest setting up two peerings between dual-stack routers:
  - One for IPv4 routing using IPv4 transport
  - Another for IPv6 routing using IPv6 transport

# IOS BGP configuration - example

```
router bgp 64066
  no bgp default ipv4-unicast
  neighbor 192.0.2.240 remote-as 64064
  neighbor 192.0.2.240 update-source Loopback0
  neighbor 192.0.2.240 version 4
  neighbor 2001:db8::1:1 remote-as 64064
  neighbor 2001:db8::1:1 update-source Loopback0
  neighbor 2001:db8::1:1 version 4
  !
  address family ipv4 unicast
  neighbor 192.0.2.240 activate
  ...
  exit-address-family
  !
  address family ipv6 unicast
  neighbor 2001:db8::1:1 activate
  network 2001:db8::/32
  exit-address-family
  !
```

# JUNOS BGP configuration

- We recommend two peerings
  - One for IPv4 and one for IPv6, just like IOS
    - Simplifies management
    - Allows external peerings to be split on logical interfaces if required
- A BGP peering is created within the protocols branch of the JUNOS CLI
  - i.e. [edit protocols bgp]
- When creating a bgp group you can specify the protocol family and optionally unicast/multicast
  - For example: 'set family inet6 multicast'

# JUNOS BGP configuration - example

```
routing-options {
  autonomous-system 64066;
}
protocols {
  bgp {
    group <name> {
      type external;
      peer-as 64064;
      neighbor 2001:db8::1:1;
      description "description";
      export [ policy1 policy2 policy3 ];
    }
  }
}
```

(policies used to handle filtering of traffic/advertisements - note order matters)

# Filtering out unwanted routes

- How do we apply policy?
- We will try to implement the following liberal policy
  - `::/8` to be denied (loopback, v4-mapped etc)
  - Deny `FE00::/9` and `FF00::/8` (link-local, multicast etc)
  - Deny `2001:db8::/32` (documentation prefix)
  - Only permit `2002::/16` from 6to4
  - Allow all others with maximum prefix length 48
- We do this by applying a prefix list
- In reality, our policies will likely be rather tighter

# IOS prefix list example

```
router bgp 64066
...
address family ipv6 unicast
...
neighbor 2001:db8::1:1 activate
neighbor 2001:db8::1:1 prefix-list ipv6-unwanted in
...
!
ipv6 prefix-list ipv6-unwanted deny 0000::/8 le 128
ipv6 prefix-list ipv6-unwanted deny fe00::/9 le 128
ipv6 prefix-list ipv6-unwanted deny ff00::/8 le 128
ipv6 prefix-list ipv6-unwanted deny 2001:db8::/32 le 128
ipv6 prefix-list ipv6-unwanted permit 2002::/16
ipv6 prefix-list ipv6-unwanted deny 2002::/16 le 128
ipv6 prefix-list ipv6-unwanted permit ::/0 le 48
ipv6 prefix-list ipv6-unwanted deny ::/0 le 128
```

# JUNOS prefix list example

```
policy-statement ipv6-bgp {
  from {
    family inet6;
    router-filter 2001:db8::/32 orlonger;
    route-filter 2002::/16 orlonger;
    route-filter 0000::/8;
    route-filter fe00::/9 orlonger;
    route-filter ff00::/8 orlonger;
    route-filter ::/0 orlonger;
  }
  then reject;
}

protocols {
  bgp {
    group <name> {
      export ipv6-bgp;
      ...
    }
    ...
  }
}
```

# Checking BGP status in IOS

- Listing BGP table
  - `show bgp ipv6 unicast`
- Listing set of peerings and status
  - `show bgp ipv6 unicast summary`
- Checking a particular peering
  - `show bgp ipv6 unicast neighbors <address>`
- Checking what we receive from a particular peer
  - `show bgp ipv6 unicast neighbors <address> received-routes`
- Checking what we send to a particular peer
  - `show bgp ipv6 unicast neighbors <address> advertised-routes`
- Note that for all the above we can use ipv4 instead of ipv6, and multicast instead of unicast

# Checking BGP status in JUNOS

- Listing BGP table
  - `show bgp summary`
- Listing set of peerings and status
  - `show bgp neighbor`
- Checking a particular peering
  - `show bgp neighbor <addr>`
- Checking what we receive from the particular peer
  - `show route receive-protocol bgp <addr>`
- Checking what we send to a particular peer
  - `show route advertising-protocol bgp <addr>`
- Note that for all the above both IPv4 and IPv6 information is displayed together

# References

- Implementing Multiprotocol BGP for IPv6 on IOS
  - [http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/products\\_configuration\\_guide\\_chapter09186a00801d65f7.html](http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/products_configuration_guide_chapter09186a00801d65f7.html)
- Configuring BGP on JUNOS
  - <http://www.juniper.net/techpubs/software/junos/junos80/swconfig80-routing/download/bgp-config.pdf>
  - or
  - <http://www.juniper.net/techpubs/software/junos/junos80/swconfig80-routing/frameset.htm>
- For BGP in general, see Configuring BGP
  - [http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products\\_configuration\\_guide\\_chapter09186a00800ca763.html](http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca763.html)