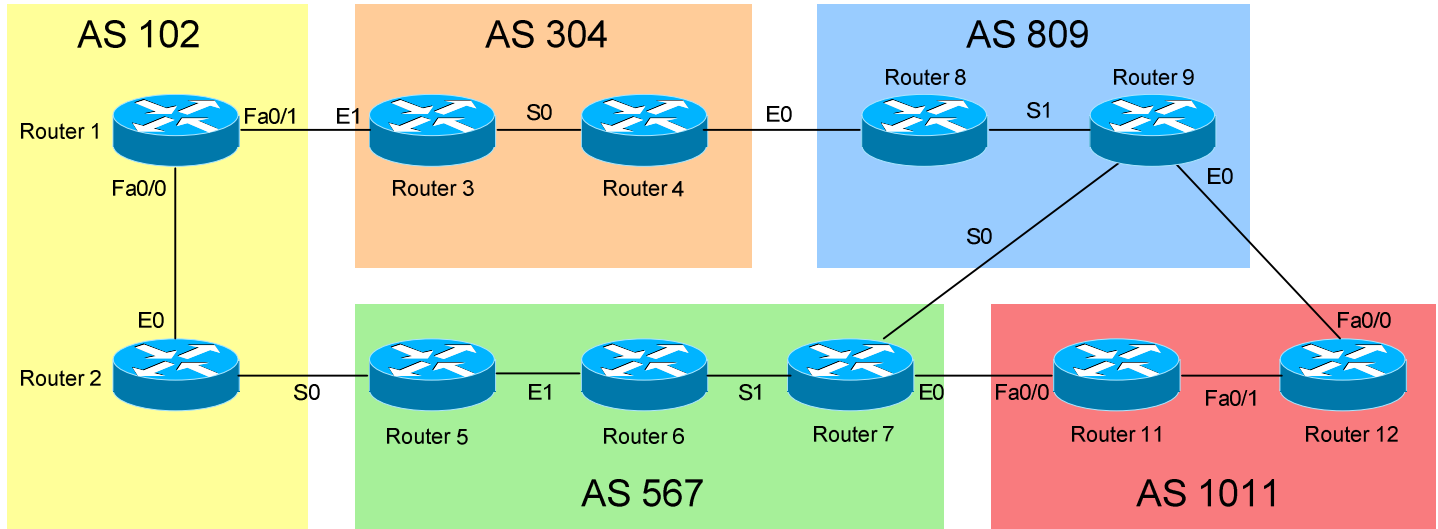


- Multi-Protocol Lab 3 Walkthrough -

Multi-Protocol Environment - Lab 3 Walkthrough



Basic Objectives:

1. Configure and cable the Serial/Ethernet interfaces as indicated in the above diagram.
2. Configure the IP addresses on the routers using the following schemes:

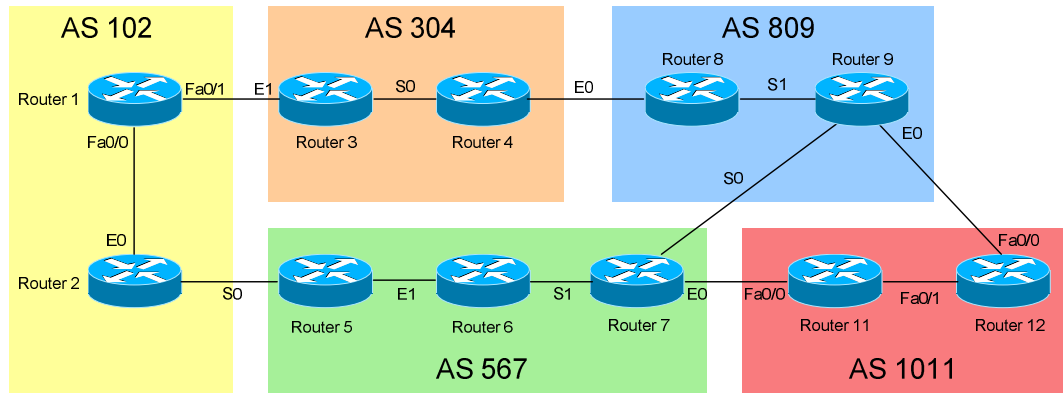
Router 1 – 2 = 192.168.12.x/24	Router 6 – 7 = 172.16.67.x/24
Router 1 – 3 = 192.168.13.x/24	Router 7 – 9 = 172.16.79.x/24
Router 3 – 4 = 192.168.34.x/24	Router 8 – 9 = 172.16.89.x/24
Router 2 – 5 = 192.168.25.x/24	Router 7 - 11 = 172.16.117.x/24
Router 4 – 8 = 192.168.48.x/24	Router 9 – 12 = 172.16.129.x/24
Router 5 – 6 = 192.168.56.x/24	Router 11 – 12 = 172.16.121.x/24

3. Configure a loopback0 interface on each router. The interface should have an address using the following scheme: Y.Y.Y.Y/24. For example, Router 4's loopback0 should be 4.4.4.4/24.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Core Objectives:**

4. All routers in the above diagram should run BGP, with the exception of Router 6. Autonomous System numbers should be configured as shown. Additionally:
 - a. All BGP updates should be sourced from the loopback interface
 - b. All BGP neighbors should be authenticated with a password of “CISCO”

We'll look at Router 2 as our example to complete this. To peer with Router 5:

```

Router2(config)# ip route 5.5.5.0 255.255.255.0 192.168.25.5

Router2(config)# router bgp 102
Router2(config-router)# neighbor 5.5.5.5 remote-as 567
Router2(config-router)# neighbor 5.5.5.5 update-source lo0
Router2(config-router)# neighbor 5.5.5.5 ebgp-multihop 2
Router2(config-router)# neighbor 5.5.5.5 password CISCO
  
```

The static route is necessary so we can reach Router 5's loopback address. Also, because we are sourcing our updates from the loopback interfaces, we must include the **ebgp-multihop** command, as our neighbors are now two hops away.

To peer with Router 1:

```

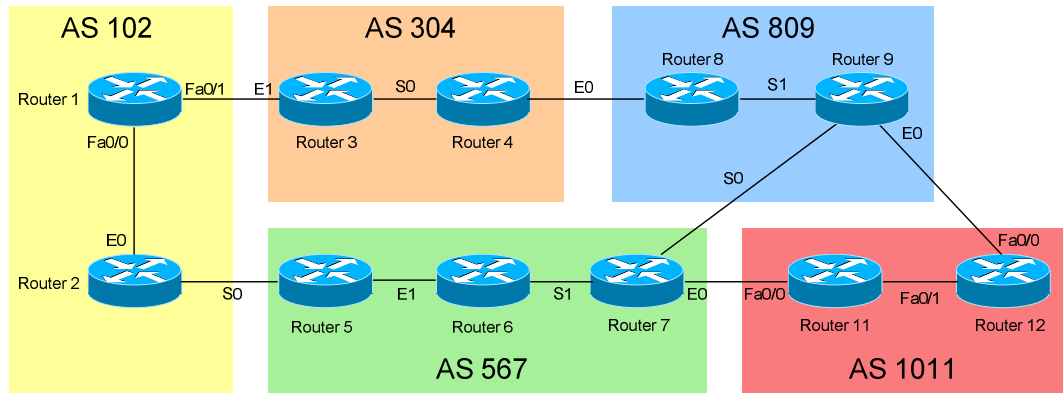
Router2(config)# router bgp 102
Router2(config-router)# neighbor 1.1.1.1 remote-as 102
Router2(config-router)# neighbor 1.1.1.1 update-source lo0
Router2(config-router)# neighbor 1.1.1.1 password CISCO
  
```

The static route is **not** necessary when neighboring with Router 1, as Router 2 will learn about the loopback network via the IGP. Also, since Router 1 is an iBGP peer, the **ebgp-multihop** command is not necessary.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Core Objectives:**

- The internal networks within each AS can utilize an IGP of your choosing. Remember, Router 6 cannot run BGP. All IGP updates should be authenticated using the most secure method available.

We'll look at Router 8 as our example to complete this. A specific IGP routing protocol was not specified, so we'll choose EIGRP.

```

Router8(config)# router eigrp 809
Router8(config-router)# no auto-summary
Router8(config-router)# network 8.0.0.0 0.255.255.255
Router8(config-router)# network 172.16.89.0 0.0.0.255
Router8(config-router)# passive-interface ethernet0

Router8(config)# key chain MYCHAIN
Router8(config-keychain)# key 1
Router8(config-keychain-key)# key-string CISCO

Router8(config)# interface s1
Router8(config-if)# ip authentication key-chain eigrp 809 MYCHAIN
Router8(config-if)# ip authentication mode eigrp 809 md5

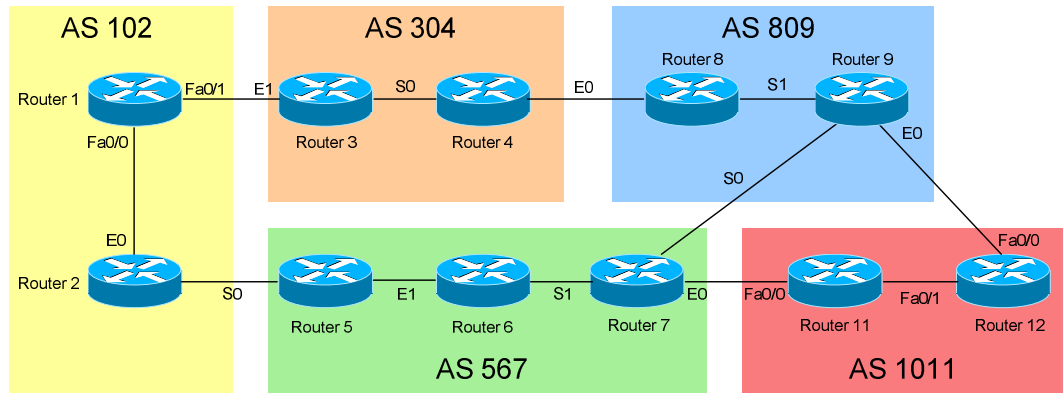
```

We made the ethernet0 interface passive, as we don't want IGP updates sent outside the autonomous system. Authentication parameters are setup using a key chain.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Core Objectives:**

6. Router 1, Router 2, Router 4, Router 6, and Router 12 should inject additional routes into this internetwork. These networks should be reachable via ICMP. You are not allowed to use static routes. The additional networks should be as followed:
 - a. Router 1 = 101.101.1-7.x/24
 - b. Router 2 = 102.102.1-7.x/24
 - c. Router 4 = 104.104.1-7.x/24
 - d. Router 6 = 106.106.1-7.x/24
 - e. Router 12 = 122.122.1-7.x/24

We'll look at Router 4 as our example to complete this. First, we'll manufacture these fake networks using loopback interfaces:

```
Router4(config)# interface lo1
Router4(config-if)# ip address 104.104.1.1 255.255.255.0
```

```
Router4(config)# interface lo2
Router4(config-if)# ip address 104.104.2.1 255.255.255.0
```

And etcetera. Make sure to create loopback interfaces for networks 104.104.1-7.x. Next, we must inject these routes into BGP:

```
Router4(config)# route-map LOOPBACKS permit 10
Router4(config-route-map)# match interface lo1 lo2 lo3 lo4 lo5 lo6 lo7
```

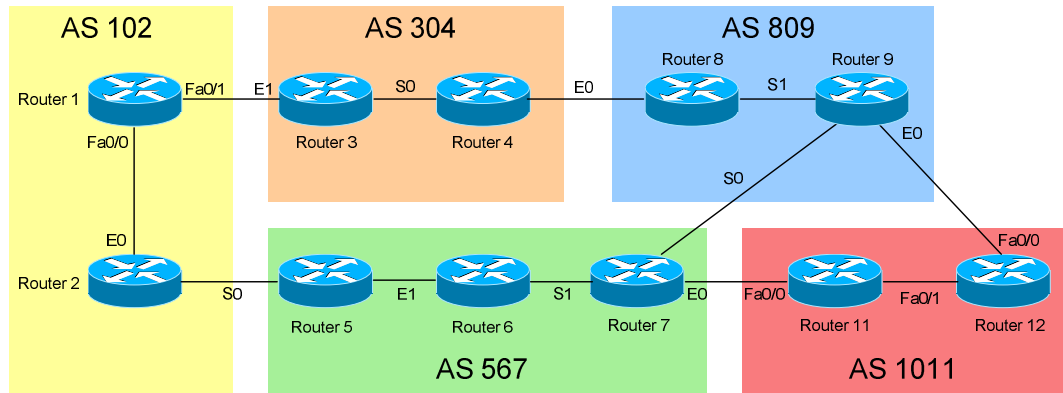
```
Router4(config)# router bgp 304
Router4(config-router)# redistribute connected route-map LOOPBACKS
```

Instead of redistributing these connected routes into BGP, we could have used individual **network** statements under the BGP process.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Core Objectives:**

7. BGP should not auto-summarize. Only the networks listed in Objective #6 from Router 4 and Router 12 should be summarized when sent outside the local AS.

To disable auto summarization, we'll look at Router 5 as an example:

```
Router5(config)# router bgp 567
Router5(config-router)# no auto-summary
```

To summarize specific networks, we'll look at Router 12 as an example:

```
Router12(config)# router bgp 1011
Router12(config-router)# no auto-summary
Router12(config-router)# aggregate-address 122.122.0.0 255.255.248.0 summary-only
```

If you do not include the **summary-only** parameter at the end of the **aggregate-address** command, BGP will send both the summary route, AND the individual routes.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Core Objectives:**

8. Ensure reachability to all routes throughout the internetwork.

This requires that all routes be injected into BGP. We'll look at Router 8 as our example again. We can either redistribute the local IGP into BGP:

```
Router8(config)# router bgp 809
Router8(config-router)# redistribute eigrp 809
```

*Or we can manually add each route into BGP using the **network** command:*

```
Router8(config)# router bgp 809
Router8(config-router)# network 8.8.8.0 mask 255.255.255.0
Router8(config-router)# network 172.16.89.0 mask 255.255.255.0
```

Or we can redistribute all connected interfaces:

```
Router8(config)# router bgp 809
Router8(config-router)# redistribute connected
```

AS567 gets a little tricky - Router 6 does not run BGP. Thus, BGP routes must be redistributed INTO the local IGP as well on both of AS 567's BGP routers (Router 5 and Router 7). Otherwise, Router 6 becomes a black hole for any packets trying to traverse AS 567. Looking at Router 5, and assuming we're using EIGRP in AS 567:

```
Router5(config)# router bgp 567
Router5(config-router)# redistribute eigrp 567
```

```
Router5(config)# router eigrp 567
Router5(config-router)# redistribute bgp 567 metric 10000 1000 255 1 1500
```

We're not done yet with this objective. When a BGP router sends a route to an eBGP peer, it specifies itself as the next-hop to reach that route. When that eBGP peer then forwards that route to an iBGP peer, it doesn't change the next-hop value. Using Router 2 as an example:

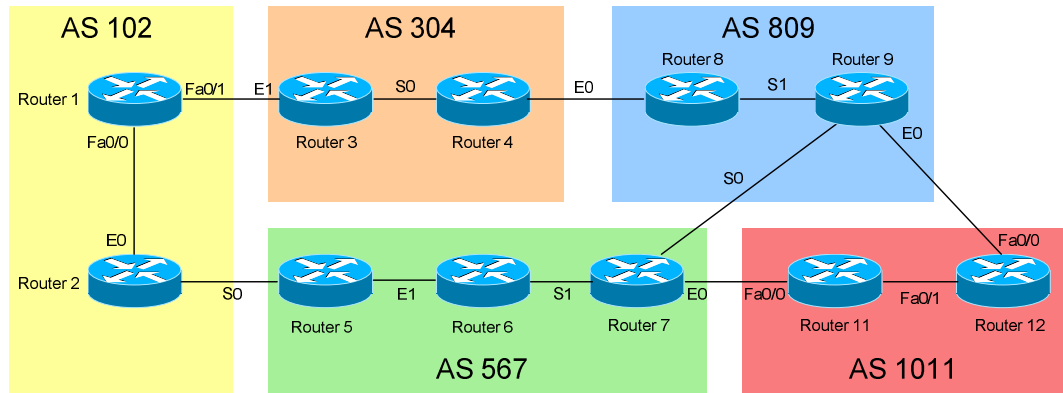
```
Router2(config)# router bgp 102
Router2(config-router)# neighbor 1.1.1.1 next-hop-self
```

When Router 2 sends an external route to Router 1, it will now include itself as the next-hop to reach that network, instead of Router 5. This command is only necessary with iBGP neighbors.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Absurd Objectives:**

- AS 1011 should prefer the path through AS 809 and AS 304 to reach AS 102, instead of through AS 567.

We have three key ways to accomplish this:

- Using the Local Preference attribute
- Using the AS-Path attribute
- Adjusting the Next-Hop

We'll use Router 12 as our example, and adjust the Local Preference to accomplish this objective:

```
Router12(config)# ip prefix-list AS102 permit 192.168.12.0/24
Router12(config)# ip prefix-list AS102 permit 1.1.1.0/24
Router12(config)# ip prefix-list AS102 permit 2.2.2.0/24
```

```
Router12(config)# route-map MATCH102 permit 10
Router12(config-route-map)# match ip address prefix-list AS102
Router12(config-route-map)# set local-preference 500
Router12(config-route-map)# route-map MATCH102 permit 20
```

```
Router12(config)# router bgp 1011
Router12(config-router)# neighbor 9.9.9.9 route-map MATCH102 in
```

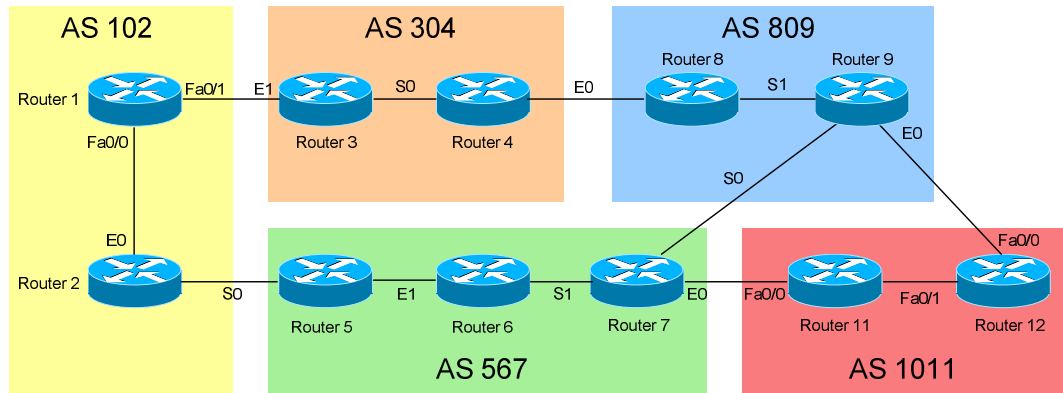
We match all the networks in AS 102 using a **prefix-list**. Next, we reference this prefix-list using a **route-map**, and then set the **local-preference** to something higher than the default of 100. Remember, a higher local-preference is preferred. Then, we apply this route-map to incoming updates from Router 9.

The local-preference attribute is sent between iBGP peers, thus Router 12 will pass this information to Router 11. Because the path through AS809 and AS 304 has a higher local-preference now, it will be preferred. We could not use the **weight** attribute to accomplish this, because weight is only locally significant.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Absurd Objectives:**

10. AS 304 should prefer the path through AS 102 and AS 567 to reach AS 1011, instead of through AS 809.

Same situation as Objective #9. We'll use Router 4 as our example, and adjust the AS-Path attribute to accomplish this:

```
Router4(config)# ip prefix-list AS1011 permit 172.16.121.0/24
Router4(config)# ip prefix-list AS1011 permit 11.11.11.0/24
Router4(config)# ip prefix-list AS1011 permit 12.12.12.0/24
```

```
Router4(config)# route-map MATCH1011 permit 10
Router4(config-route-map)# match ip address prefix-list AS1011
Router4(config-route-map)# set as-path prepend 100 100 100 100
Router4(config-route-map)# route-map MATCH1011 permit 20
```

```
Router4(config)# router bgp 304
Router4(config-router)# neighbor 8.8.8.8 route-map MATCH1011 in
```

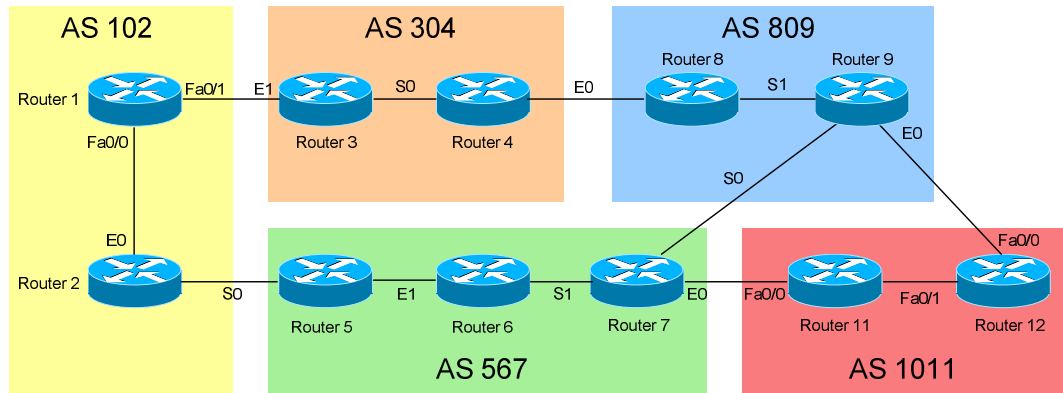
We match all the networks in AS 1011 using a **prefix-list**. Next, we reference this prefix-list using a **route-map**, and then we **prepend** a “slew” of fake AS's to the AS path. Remember, the shortest AS path is preferred; thus, by lengthening the AS path through AS 809, it becomes less preferred.

Finally, we apply this route-map to incoming updates from Router 8.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Absurd Objectives:**

11. Router 11 should not advertise any routes to Router 12 sourced from AS 102. Do not use an IP access-list or prefix-list to accomplish this.

This configuration is completed on Router 11.

```

Router11(config)# ip as-path access-list 1 permit _102$

Router11(config)# route-map DENY102 deny 10
Router11(config-route-map)# match as-path 1
Router11(config-route-map)# route-map DENY102 permit 20

Router11(config)# router bgp 1011
Router11(config-router)# neighbor 12.12.12.12 route-map DENY102 out

```

*We're not allowed to use an IP access-list, so instead we'll use an as-path access-list. The regular expression **_102\$** matches any routes originated in AS102.*

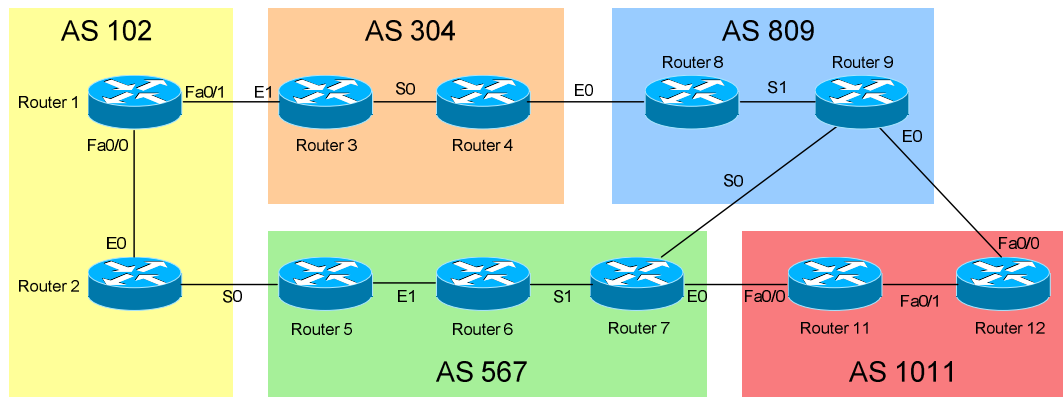
*We match this using a route-map, and **deny** the routes matched in this as-path access-list. Don't forget to include the final permit statement in the route-map, otherwise all routes will be denied.*

Finally, we apply this route-map to outgoing updates to Router 12.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Absurd Objectives:**

12. At AS borders, BGP peers should filter outgoing updates to *not* include loopback0 networks of internal routers.

We'll use Router 7 as our example:

```

Router7(config)# ip prefix-list AS567 permit 5.5.5.0/24
Router7(config)# ip prefix-list AS567 permit 6.6.6.0/24
Router7(config)# ip prefix-list AS567 permit 7.7.7.0/24

Router7(config)# route-map NOLOOPBACKS deny 10
Router7(config-route-map)# match ip address prefix-list AS567
Router7(config-route-map)# route-map NOLOOPBACKS permit 20

Router7(config)# router bgp 567
Router7(config-router)# neighbor 9.9.9.9 route-map NOLOOPBACKS out
Router7(config-router)# neighbor 11.11.11.11 route-map NOLOOPBACKS out

```

We match all the loopback routes in AS 567 using a prefix-list. We then match (and deny) those loopback routes using a route-map. Finally, we apply the route-map to outgoing updates to Router 9 and Router 11.

13. Each BGP router should be configured to limit the number of routes received by a neighbor to 1000.

The easiest objective in this lab. Only one command is necessary per neighbor. Using Router 3 as our example:

```

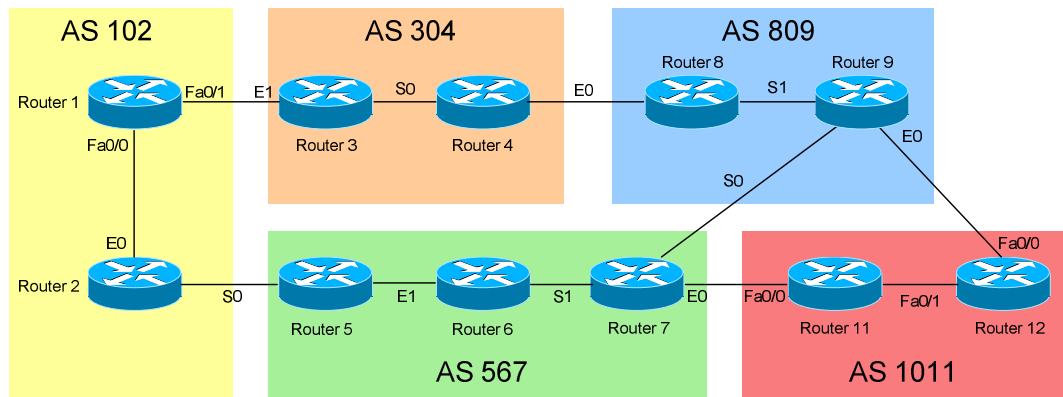
Router3(config)# router bgp 304
Router3(config-router)# neighbor 1.1.1.1 maximum-prefix 1000
Router3(config-router)# neighbor 4.4.4.4 maximum-prefix 1000

```

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Absurd Objectives:**

14. Ensure each router is configurable by telnet:
- Use a local username of “silent” and a password of “genius”
 - Routers should be ping-able by name by all other routers.
 - Devices within the same AS should be able to access their routers via telnet at all times
 - Devices from another AS should be able to access remote routers only during working hours (8:00a.m.-5:00p.m.) on weekdays.

We'll use Router 9 as our example:

```
Router9(config)# username silent password genius
```

```
Router9(config)# ip host Router1 192.168.12.1
```

```
Router9(config)# ip host Router2 192.168.12.2
```

And etcetera. Make sure you create a host entry for each Router. We've filtered the loopback interfaces, so you can't point to that address.

```
Router9(config)# time-range WORKINGHOURS
```

```
Router9(config-time-range)# periodic weekdays 08:00 to 17:00
```

```
Router9(config)# access-list 101 permit ip 172.16.89.0 0.0.0.255 any
```

```
Router9(config)# access-list 101 permit ip any any time-range WORKINGHOURS
```

```
Router9(config)# line vty 0 4
```

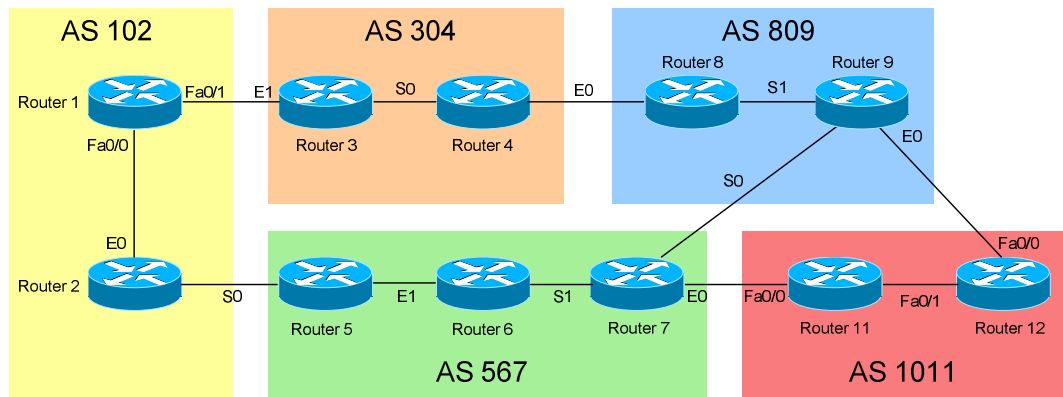
```
Router9(config-line)# login local
```

```
Router9(config-line)# access-class 101 in
```

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Really Absurd Objectives:**

15. AS 567 is toying with the idea of using IPv6. The interfaces connecting Router 5, Router 6, and Router 7 should be configured with an IPv6 address. Use an Aggregate Global addressing scheme, with a /96 network prefix. Ensure the routers can ping each other's directly connected interfaces using IPv6.

First, we'll use Router 5 as our example:

```

Router5(config)# ipv6 unicast-routing

Router5(config)# interface ethernet1
Router5(config-if)# ipv6 address 2001:1111::1/96

```

Next, looking at Router 6:

```

Router6(config)# ipv6 unicast-routing

Router6(config)# interface ethernet1
Router6(config-if)# ipv6 address 2001:1111::2/96

Router6(config)# interface serial1
Router6(config-if)# ipv6 address 2001:2222::1/96

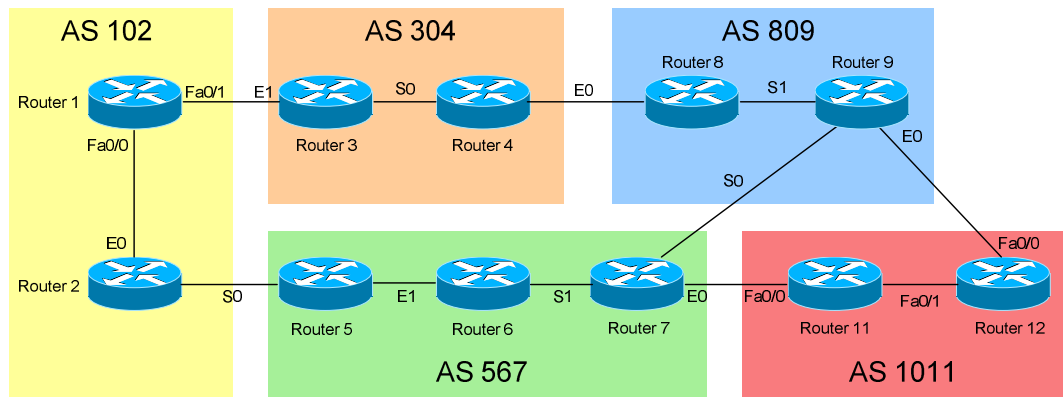
```

IPv6 must first be globally enabled on the router. I arbitrarily chose the 2001.1111::/96 and 2001:2222::/96 aggregate global networks. This is the only configuration necessary to complete this objective.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**Really Absurd Objectives:**

16. The network between Router 1 and Router 2 contains a multicast server, supporting a multicast group at address 229.1.1.150:
- Ensure reach-ability to this group from all interfaces on Router 1, Router 2, Router 3, Router 4, and Router 9.
 - All interfaces on these routers should join this multicast group.
 - Use the least bandwidth-intensive mode for whatever multicast routing protocol you choose.
 - Do not use the default version of the multicast group management protocol on each interface.

We'll look at Router 1 as our example:

```

Router1(config)# ip multicast-routing
Router1(config)# ip pim rp-address 192.168.12.1

Router1(config)# interface fa0/0
Router1(config-if)# ip igmp version 1
Router1(config-if)# ip igmp join-group 229.1.1.150
Router1(config-if)# ip pim sparse-mode

Router1(config)# interface fa0/1
Router1(config-if)# ip igmp version 1
Router1(config-if)# ip igmp join-group 229.1.1.150
Router1(config-if)# ip pim sparse-mode

```

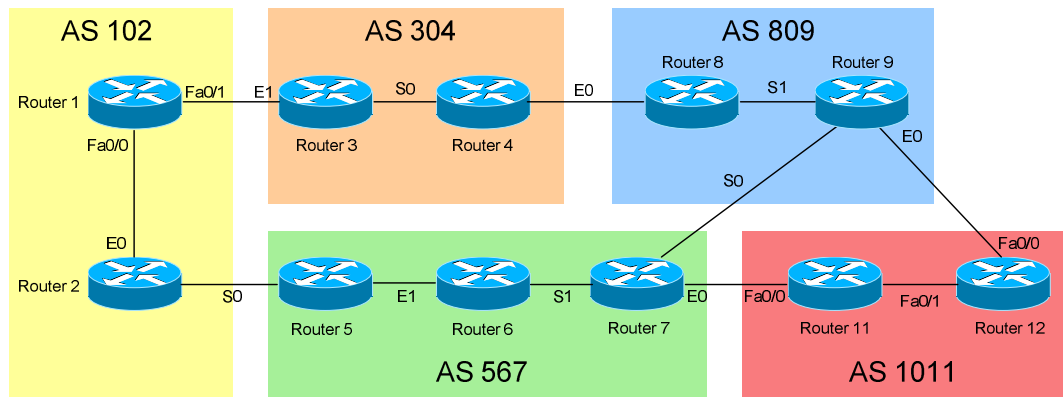
IGMPv2 is the default version, so we manually specified version 1. We also forced each interface to join the multicast group. PIM Sparse-Mode is the least bandwidth intensive mode for PIM.

One of the routers had to become the Rendezvous Point (RP). We used the **ip pim rp-address** command to accomplish this. This has to be configured on every router, including the RP itself. I arbitrarily chose Router 1 to be the RP.

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Multi-Protocol Environment - Lab 3 Walkthrough (continued)**The Absurdly Absurd “What-Is-He-Thinking” Objective:**

17. Accomplish objective #16 *without any* multicast configuration on Router 8, *or any other* routers not listed in objective #16.

This necessitates that we create a tunnel between Router 4 and Router 9. On Router 4:

```

Router4(config)# ip multicast-routing
Router4(config)# ip pim rp-address 192.168.12.1

Router4(config)# interface tunnel0
Router4(config-if)# ip unnumbered
Router4(config-if)# ip igmp join-group 229.1.1.150
Router4(config-if)# ip pim sparse-mode
Router4(config-if)# tunnel source ethernet0
Router4(config-if)# tunnel destination 172.16.89.9

```

On Router 9:

```

Router9(config)# ip multicast-routing
Router9(config)# ip pim rp-address 192.168.12.1

Router9(config)# interface tunnel0
Router9(config-if)# ip unnumbered
Router9(config-if)# ip igmp join-group 229.1.1.150
Router9(config-if)# ip pim sparse-mode
Router9(config-if)# tunnel source serial1
Router9(config-if)# tunnel destination 192.168.48.4

Router9(config)# ip mroute 0.0.0.0 0.0.0.0 tunnel0

```

* * *

All original material copyright © 2009 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.