# IP Routing the Internet - BGP

In Lab 2, we began exploring routing between groups of local routers. In this lab, we look at connecting routers through an internet connection. Of course, we cannot possibly model the entire Internet, but we can look at specific parts and how they interact with one another. We’ll need to make some significant simplifications along the way.

# Internet Service Provider – Subscriber Services

Eonlink[[1]](#footnote-1), a new telecom has moved into Boatsville. To model this, we will need to build a local subscriber loop in Boatsville:

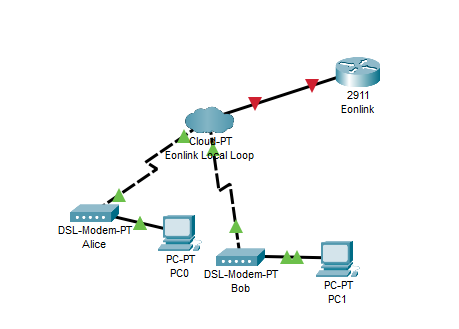
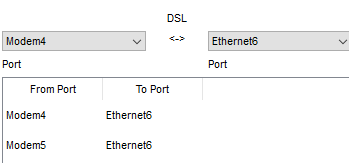
1. Place a PT-Cloud onto the canvas
2. Rename this cloud to be “Eonlink Local Loop”
3. Place two DSL modems onto the canvas
   1. Rename one of the DSL modems to “Alice” and the other to “Bob”
   2. Connect Alice to the PT-Cloud using the “Modem4” port

Figure 1 - EonLink's Local Loop

* 1. Connect Bob to the PT-Cloud using the “Modem5” port

1. Alice and Bob both have exactly one PC, so drag two PCs onto the canvas, and connect one to each of the modems.
2. Add a 2911 router which will serve as Eonlink’s local router. Change its name to “Eonlink”
3. Connect the Eonlink Local Loop to Eonlink’s router
4. Next, we need to direct the DSL traffic from the cloud to the router. To do this, click on the cloud, and add a linkage from the two modem ports to the Ethernet6 interface that goes to the 2911 router.
5. The EonLink router needs some configuration. Configure the ethernet adapter Gig0/0 have an IP address of 160.54.1.1/24, and turn it on.
6. EonLink’s customers have to configure their own IP addresses. Both use the same gateway: 160.54.1.1, and then each PC gets an IP:
   1. Alice 160.54.1.15
   2. Bob 160.54.1.30
7. Ping from both PCs to the router, fix any errors that may have occurred.
8. Can the PC’s ping each other? Would you want them to?

# Internet Service Provider – Corporate

Bombast markets themselves as provider of hyperbolic IP services with direct-to-the-frontdoor fiber optic cables. Their infrastructure enables direct connections to large datacenters, including Microsloth the number one seller of baby 2- and 3-toed south American sloths; and Fakebook, a service that distributes fake news 24x7, and Factorial70!, an on-line advertising company that happens to provide other services.

1. Place a PT-Empty router onto the canvas, rename it “Bombast”
2. Cut its power, drop in four PT-ROUTER-NM-1FGE modules into the left-most slots, and a PT-ROUTER-NM-1CGE module into the right most slot. Turn it back on.
3. Add three copies of a PT-Empty router, each with a single PT-ROUTER-NM-1FGE (copy and paste works great here).
4. Use the orange fiber-optic cable to connect the routers, it might be handy to fill in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| From | Interface | To | Interface |
| Bombast |  | Microsloth |  |
| Bombast |  | Fakebook |  |
| Bombast |  | Factorial70! |  |

## Corporate IP Address Ranges

Larger organizations register for and port their own IP addresses. In this case, these three companies each maintain the following list of subnets. The routers represent a much more complicated local network of networks so the one router represents an entire corporate network infrastructure:

|  |  |  |
| --- | --- | --- |
| Company | Interface | Subnets |
| Microsloth | Gig0/0 | 216.253.57.0/24 |
| Microsloth | Gig1/0 | 199.246.74.0/24 |
| Microsloth | Gig2/0 | 211.210.235.0/24 |
| Fakebook | Gig0/0 | 216.178.151.0/24 |
| Fakebook | Gig1/0 | 200.172.126.0/24 |
| Fakebook | Gig2/0 | 220.238.220.0/24 |
| Fakebook | Gig3/0 | 207.188.197.0/24 |
| Fakebook | Gig4/0 | 214.182.166.0/24 |
| Fakebook | Gig5/0 | 217.249.74.0/24 |
| Factorial70! | Gig0/0 | 164.14.0.0/16 |

1. Configure each router with one PT-ROUTER-NM-1CGE for each subnet it supports, but don’t remove the single fiber “1FGE” card that was added in the previous step.
2. Next, hang a server from each of the three corporate providers, the servers according to the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| Company | Interface | IP Address | Gateway |
| Microsloth | Gig1/0 | 199.246.74.80 | 199.246.74.1 |
| Fakebook | Gig4/0 | 207.188.197.75 | 207.188.197.1 |
| Factorial70! | Gig0/0 | 164.14.18.24 | 164.14.18.1 |

1. Configure the interfaces on each of the routers, and bring them on-line. The CPE routers’ interfaces should all be the subnet, “.1” (e.g. Microsoth’s Gig0/0 should be 216.253.57.1).

## Private Router Links

The private IP addresses (e.g. 10.0.0.0/16 and 192.168.0.0/24) are really useful for addressing private / link-local addresses. In this case, we already used the 10.0.0.0/16 for Eonlink, but that doesn’t mean Bombast can’t use it too – those addresses will *never* propagate over the internet, so both companies can use them but never advertise them.

Configure the links (on both sides) between Bombast’s router and each of the companies’ routers. The links are shown below. Don’t forget to turn them on:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Destination | Net | Bombast Side | | Customer Side | |
| Interface | IP | Interface | IP |
| Microsloth | 10.1.1.0/24 | Gig9/0 | 10.1.1.1 | Gig9/0 | 10.1.1.2 |
| Fakebook | 10.2.1.0/24 | Gig8/0 | 10.2.1.1 | Gig9/0 | 10.2.1.2 |
| Factorial70! | 10.3.1.0/24 | Gig7/0 | 10.3.1.1 | Gig9/0 | 10.3.1.2 |

After configuring these links – its time to verify connectivity – should **should** be able to ping each of the routers using their private (10.x.x.x) address. You **should not** be able to ping “through” the routers to the servers. We haven’t done that. Yet.

## Inter-Corporate Routing - Static

All three of Bombast’s companies have public facing IP addresses and would expect to have the capability to run servers on those addresses, and maybe even reach other – for example, every corporate customer probably wants to use Factorial70!’s on-line advertising and other services. So, Bombast could start entering routes for each of their customers into their router’s *static* routing table.

And while we could, this will quickly become a problem. When Eonlink and Bombast want to meet at an *exchange point*, then Eonlink will need to update all of their routers to include all of Bombast’s routes, and likewise, Bombast will need to manually update their routers to include Eonlink’s customers. Every time there is a change, a net admin from one company will need to call the other to update their routes again. Painful!

So, while I’m not immune from tasking with some terribly tedious task, lets assume that this is just too painful even for me to assign, and move on to BGP!

## Border Gateway Protocol

BGP is designed to provide routing links between organizations. In this strategy, each company is responsible for deciding how to route on their networks and only their networks. BGP will allow the internet service providers to advertise routes to their customers in a well-controlled and (mostly) secure fashion. BGP takes two forms: internal and external. In this example, we’ll only worry about *external* BGP.

In lecture, we talked about *autonomous system numbers*. Since these providers & customers are large enough, they have their own AS numbers:

|  |  |
| --- | --- |
| Company | AS Number |
| Bombast | 64520 |
| Eonlink | 64682 |
| Factorial70! | 64598 |
| Fakebook | 65012 |
| Microsloth | 65253 |

On each of the three customer routers, we need to do the following commands, adjusted for the specifics of each router (example shown is for Factorial70!):

1. Enable BGP:  
   Router(config)# router bgp **64598**
2. Define Neighbor AS (repeat for each *directly connected neighbor*):  
   Router(config-router)# neighbor **10.3.1.1** remote-as 64520
3. Select Routes to advertise (list each of the subnets on each customer):  
   Router(config-router)# network **164.14.18.24**

The Bombast router needs to have BGP enabled as well, but it doesn’t have any local routes, it will only broadcast the routes its learned from its customers. Configure this one as well.

## Internet Exchanges

Both Bombast and Eonlink decide to join an internet exchange point. For this to happen, we need to model the exchanges router.

1. Place a PT-EMPTY onto the canvas
2. Change its display name to be “Exchange”
3. Turn off the router
4. Place three PT-ROUTER-NM-FGE cards into the router
5. … and turn it back on again.
6. Use the fiber optic connection to connect the two networks.

**OH NO!** We don’t have an available fiber connection in Eonlink’s router! Even worse, the router we have doesn’t support the SFP. So, we need to add another router for Eonlink. Pick another PT-EMPTY, and use that as the “border router” for the Eonlink network.

First, configure the link between the old Eonlink router and the new border router:

1. Starting with the old router, configure interface gig0/1 to have a “private” IP address of 10.10.10.1 and netmask of 255.255.255.0
2. Move to the new router, and configure the gig0/0 link to have the private IP address of “10.10.10.2”, and now that link should turn green.
3. Also on the new Eonlink router, configure the link to the new exchange, using 10.8.1.2 and a netmask of 255.255.255.0
4. Next, setup the internet exchange router, configure its interface to Eonlink as 10.8.1.1, also with netmask of 255.255.255.0
5. Setup the link from the exchange router to Bomast, using 10.4.1.2 on the exchange and 10.4.1.1 on Bombast’s side.
6. On Bombast’s router, go back to the BGP configuration (use the same route bgp 64520), and add the exchange to the neighbor. The new exchange’s AS# is 65128.
7. Verify that Bombast’s routes are appearing on Eonlink’s border router.

## One More Bridge to Cross

Eonlink’s border router needs to know about the internal links inside Eonlink’s network. This is a job for OSPF! Configure OSPF on both Eonlink routers.

1. On the Border router, turn on OSPF:  
   Router(config)# router ospf 100
2. Select network to use for OSPF:  
   Router(config-router)# network 10.10.10.1 0.0.0.255 area 0
3. Enable the redistribution of wealth BGP learned routes using the Eonlink’s AS number:  
   Router(config-router)# redistribute bgp 64682  
   And then, after that, we need to return to config mode:  
   Router(config-router)# exit
4. And then, edit the BGP routing configuration to redistribute OSPF routes:  
   Router(config)# router bgp 64682  
   Router(config-router)# redistribute ospf 100
5. On the old Eonlink router, turn on OSPF:

Router(config)# router ospf 100

1. Select the subnet to use:  
   Router(config-router)# network 10.10.10.2 0.0.0.255 area 0
2. Select to redistribute static routes:  
   Router(config-router)# redistribute static subnets  
   Wait for the message “Router#

00:33:25: %OSPF-5-ADJCHG: Process 100, Nbr 160.54.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done”

1. Verify that the internal eonlink has the BGP routes: “show ip route”:

10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

C 10.10.10.0/24 is directly connected, GigabitEthernet0/1

L 10.10.10.1/32 is directly connected, GigabitEthernet0/1

160.54.0.0/16 is variably subnetted, 2 subnets, 2 masks

C 160.54.1.0/24 is directly connected, GigabitEthernet0/0

L 160.54.1.1/32 is directly connected, GigabitEthernet0/0

O E2 199.246.74.0/24 [110/20] via 10.10.10.2, 00:01:02, GigabitEthernet0/1

O E2 207.188.197.0/24 [110/20] via 10.10.10.2, 00:01:02, GigabitEthernet0/1

1. And now, the exchange should see a similar routing table “show ip route”
2. Check pings – Alice should now be able to ping all of the servers, and even access their web services!

If you’ve reached this point, you’ve done it! You’ve modeled the ways the internet is routed. Almost. We didn’t setup a gateway of last resort. Lets give that a try now:

1. Add another PT-Cloud router, give it a 1FGE module.
2. Connect the new router to the exchange using the fiber connection.
3. Add a server called “Amazon” to the canvas
   1. Configure its IP address to be 10.5.1.2
   2. Configure its gateway to be 10.5.1.1
4. Add a DSL model to the canvas, connect it to the cloud and to the Server.
5. Like we did before, click on the cloud, then DSL, and add a map from “Modem 4” to “GigabitEthernet”.
6. Go to the exchange, configure this new link:
   1. Configure the ethernet address to be 10.5.1.1, netmask 255.255.255.0
7. And, tell the exchange that our “gateway of last resort” is the new cloud connection:
   1. Router(config)# ip default-network 10.5.1.1
8. Verify that you can ping the PC

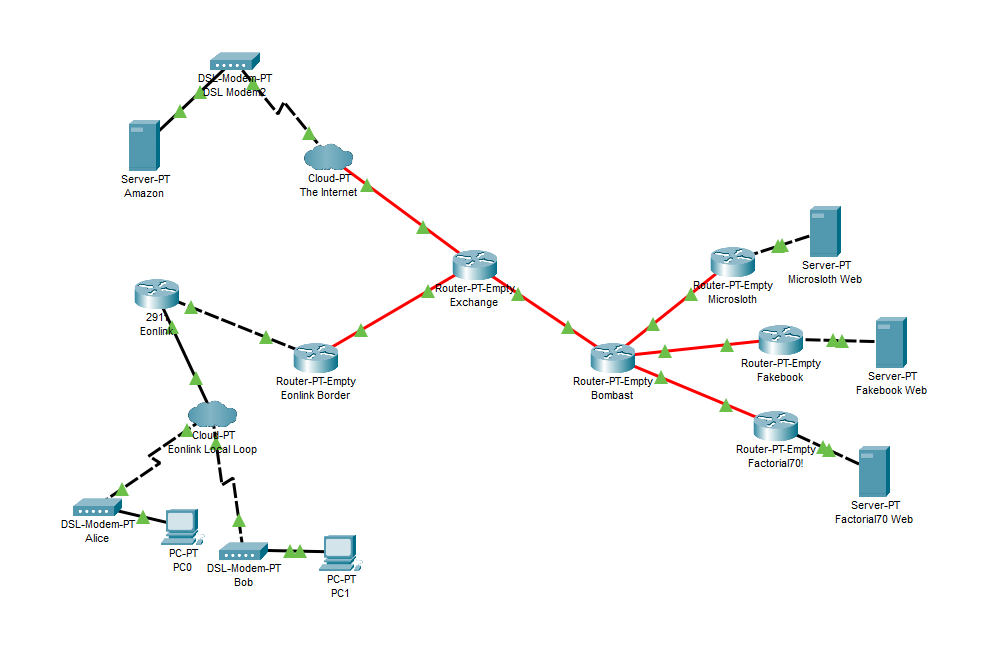
The exchange is set, but our border router isn’t! So, tell the border router that our gateway of last resort is the exchange:

1. Configure default-network on border router:  
   Router(config)# ip default-network 10.8.1.1
2. Ping the server: ping 10.5.1.2

OK, but Eonlink’s router doesn’t have a default gateway, so:

1. Configure its default:  
   Router(config)# ip default-network 10.10.10.2
2. Ping the server.

Now that Eonlink has the route, the PCs inherit it too!



1. Eonlink – Our DSL service is so slow that it takes *eons* to download data! [↑](#footnote-ref-1)