Redistribution and Filtering

Redistribution is necessary when routing protocols connect and must pass routes between the two. This can happen in a number of situations, but some examples include:

- Organizations transitioning routing protocols
- Businesses merge, and so must their networks
- OSPF or EIGRP is used at the access and distribution layer of an enterprise and BGP is used in the core

The challenge to redistributing routing protocols is that each routing protocol uses it own metric and they are not compatible with each other. Furthermore, there is no magic algorithm than can automatically translate metrics between, say RP and BGP.

To deal with this dilemma, a new seed metric is used as a staring point when redistribution is configured.

Configuring Redistribution

To configure redistribution between routing protocols, the ***redistribute*** *protocol* command is used under the routing protocol that recieves the routes.

```
R1(config-router)# redistribute protocol [AS] [metric metric-vlaue]
```

The process-id field above is the AS number for RIP, EIGRP, BGP. For OSPF, use the process ID. Also, both RIP and EIGRP require the use the metric keyword!

EIGRP Redistribution Example:

```
R1(config)# router eigrp 10
R1(config-router)# redistribute ospf 20 metric 1000 100 255 1 1500
```

The example above shows OSPF being redistruted into EIGRP with a metric of 1000 100 255 1 1500. That is a lot of different numbers for an EIGRP cost! That's because EIGRP redistribution metric requires you to input all of the metric calculation manually:

- bandwidth
- delay
- reliability
- loading
- mtu

You can perform a ***show interface*** on the outgoing router interface prior to implementing the redistribution to see what values the router is currently using.

OSPF Redistribution Example:

```
R1(config)# router ospf 100
R1(config-router)# redistribute eigrp 10 subnets
```

The example above redistributes EIGRP routes into OSPF. The subnets keyword at the end of the redistribute command is extremely important! Without this keyword, OSPF will redistribute networks at their classful boundaries – not something most administrators want.

If you don't use it the IOS will even give you a warning. Make sure to include it.

Distribute Lists
Distribute lists are access lists applied to the routing process, determining which networks are allowed into the routing table or included in updates. They essentially act as a filter.

An access list applied to routing = distribute lists
When creating a distribute list, use the following steps:
Step 1. Identify the network addresses to be filtered and create an ACL – permitting the networks you want to be advertised.
Step 2. Determine if you want to filter updates coming into the router or leaving the router.
Step 3. Assign the ACL using the distribute-list command.

Incoming Distribute Lists:

```
R1(config-router)# distribute-list {acl-number | name} in [interface-type number]
```

Outgoing Distrubute Lists:

```
R1(config-router)# distribute-list {acl-number | name} out [interface-name | routing-process |
AS-number]
```

Route Maps
When a routing update arrives at an interface, a series of steps occur to process it correctly. The diagram below outlines those steps and serves as a foundation for the rest of this route redistribution and filtering section.

Route maps are extremely flexible and are used in a number routing scenarios including:

- **Controlling redistribution** based on permit/deny statements
- **Defining policies in policy-based routing (PBR)**
- **Add more mature decision making to NAT decisions** than simply using static translations
- **When implementing BGP PBR**

Route maps allow an administrator to define specific traffic and then take automated actions against it to control how routing information is processed and forwarded. Route maps uses logic similar to if/then statements in simple scripting.
In route map terms, it *matches* traffic against conditions and *sets* options for that traffic.

NOTE: *If you have downloaded the Switch Exam Guide, you will notice the similarity between the syntax structure of route maps and VACLs.*
Each statement in a route map has a sequence number, which is read from lowest to highest. The router stops reading statements when it reaches its first matching statement.

Understand that there is an implicit deny included in all route maps. If traffic does not match any statement, it is denied.

Basic Route Map Configuration

```
R1(config)# route-map {tag} permit | deny [sequence_number]
```

That is how all route maps begin. *Permit* means that any traffic matching the *match* statement that follows is processed by the route map. Deny means that any traffic matching the *match* statement that follows is NOT processed by the route map. Know the difference.

Match & Set Conditions

If no match condition exists, the statement matches anything (similar to a ‘permit any’).
If no set condition exists, the statement is simply permitted or denied with no additional changes made.
If multiple match conditions are used on the same line, it is interpreted as a logical OR. In other words, if one condition is true, a match is made. For example, the router would interpret ‘match a b c’ as ‘a or b or c’.
If multiple match conditions are used on consecutive lines, it is interpreted as a logical AND. In other words, all conditions must be true before a match is made. For example, the router would interpret the following commands as *match a and b and c*:

```
route-map EXAMPLE permit 5
match a
match b
match c
```

Important route redistribution match conditions

ip address
Refers to an access list that permits or denies networks
ip address prefix-list
Refers to a prefix list that permits or denies prefixes
ip next-hop
Refers to an access list that permits or denies ip next hops IP addresses
ip route-source
Refers to an access list that permits or denies advertising router IP addresses
length
Permits or denies packets based on length (in bytes)
metric
Permits or denies routes with specific metrics from being redistributed
route-type
Permits or denies redistribution based on the route type listed
tag
Routes can be labeled with a number that identifies it

Important route redistribution set conditions
metric
Sets the metric for redistributed routes
tag
Tags a route with a numbered identifier

Route Map Verification
Use the ***show route-map*** command to verify route maps and PBR entries are filtering as expected.