

## - IGRP -

### IGRP (Interior Gateway Routing Protocol)

IGRP is a Cisco-proprietary Distance-Vector protocol, designed to be more scalable than RIP, its standardized counterpart.

IGRP adheres to the following Distance-Vector characteristics:

- IGRP sends out periodic routing updates (every **90 seconds**).
- IGRP sends out the full routing table every periodic update.
- IGRP uses a form of distance as its metric (in this case, a composite of **bandwidth** and **delay**).
- IGRP uses the Bellman-Ford Distance Vector algorithm to determine the best “path” to a particular destination.

Other characteristics of IGRP include:

- IGRP supports **only IP routing**.
- IGRP utilizes **IP protocol 9**.
- IGRP routes have an administrative distance of **100**.
- IGRP, by default, supports a maximum of **100 hops**. This value can be adjusted to a maximum of **255 hops**.
- IGRP is a **classful** routing protocol.

IGRP uses **Bandwidth** and **Delay of the Line**, by default, to calculate its distance metric. **Reliability**, **Load**, and **MTU** are optional attributes that can be used to calculate the distance metric.

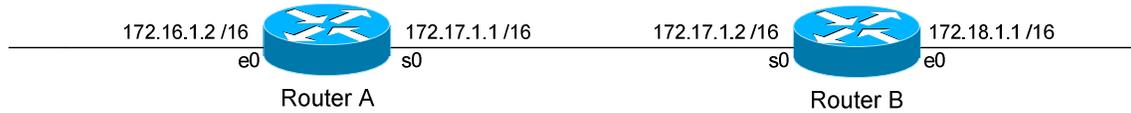
IGRP requires that you include an **Autonomous System (AS)** number in its configuration. Only routers in the same Autonomous system will send updates between each other.

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto: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>.

## Configuring IGRP



Routing protocol configuration occurs in Global Configuration mode. On Router A, to configure IGRP, we would type:

```
Router(config)# router igrp 10
Router(config-router)# network 172.16.0.0
Router(config-router)# network 172.17.0.0
```

The first command, *router igrp 10*, enables the IGRP process. The “10” indicates the Autonomous System number that we are using. Only other IGRP routers in Autonomous System 10 will share updates with this router.

The *network* statements tell IGRP which networks you wish to advertise to other RIP routers. We simply list the networks that are directly connected to our router. Notice that we specify the networks at their classful boundaries, and we do not specify a subnet mask.

To configure Router B:

```
Router(config)# router igrp 10
Router(config-router)# network 172.17.0.0
Router(config-router)# network 172.18.0.0
```

The routing table on Router A will look like:

```
RouterA# show ip route

Gateway of last resort is not set

C    172.16.0.0 is directly connected, Ethernet0
C    172.17.0.0 is directly connected, Serial0
I    172.18.0.0 [120/1] via 172.17.1.2, 00:00:00, Serial0
```

The routing table on Router B will look like:

```
RouterB# show ip route

Gateway of last resort is not set

C    172.17.0.0 is directly connected, Serial0
C    172.18.0.0 is directly connected, Ethernet0
I    172.16.0.0 [120/1] via 172.17.1.1, 00:00:00, Serial0
```

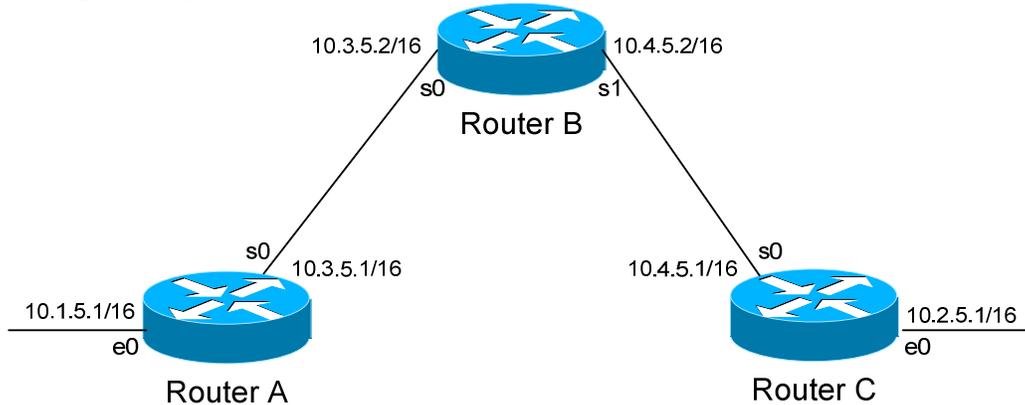
\*\*\*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto: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>.

## Limitations of IGRP

The example on the previous page works fine with IGRP, because the networks are contiguous and the subnet masks are consistent. Consider the following example:



This particular scenario will *still* work when using IGRP, despite the fact that we've subnetted the major 10.0.0.0 network. Notice that the subnets are contiguous (that is, they belong to the same major network), and use the same subnet mask.

When Router A sends an IGRP update to Router B via Serial0, it will not include the subnet mask for the 10.1.0.0 network. However, because the 10.3.0.0 network is in the same major network as the 10.1.0.0 network, it will **not summarize** the address. The route entry in the update will simply state "10.1.0.0".

Router B will accept this routing update, and realize that the interface receiving the update (Serial0) belongs to the same major network as the route entry of 10.1.0.0. It will then apply the subnet mask of its Serial0 interface to this route entry.

Router C will similarly send an entry for the 10.2.0.0 network to Router B. Router B's routing table will thus look like:

**RouterB#** *show ip route*

```

Gateway of last resort is not set

    10.0.0.0/16 is subnetted, 4 subnets
C       10.3.0.0 is directly connected, Serial0
C       10.4.0.0 is directly connected, Serial1
I       10.1.0.0 [120/1] via 10.3.5.1, 00:00:00, Serial0
I       10.2.0.0 [120/1] via 10.4.5.1, 00:00:00, Serial1
  
```

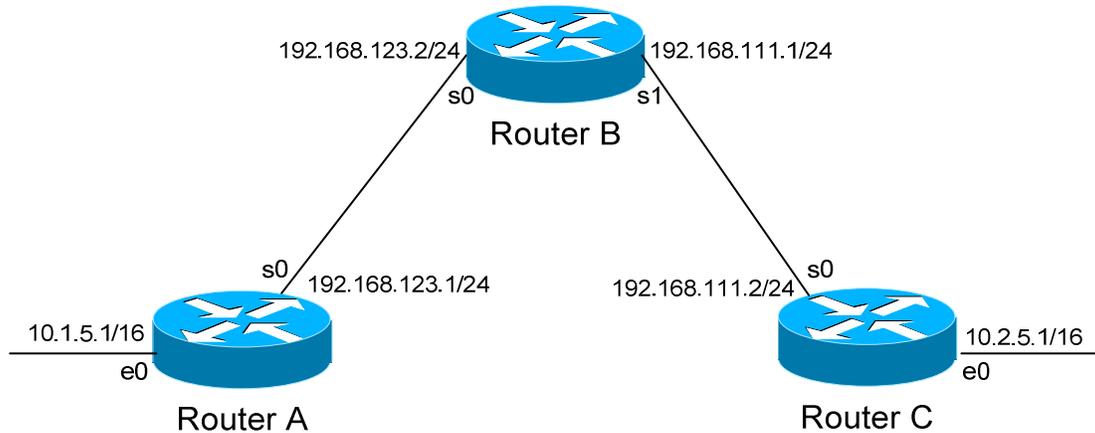
\*\*\*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto: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>.

**Limitations of IGRP (continued)**

Consider the following, slightly altered, example:



We'll assume that IGRP is configured correctly on all routers. Notice that our networks are no longer contiguous. Both Router A and Router C contain *subnets* of the 10.0.0.0 major network (10.1.0.0 and 10.2.0.0 respectively).

Separating these networks now are two Class C subnets (192.168.123.0 and 192.168.111.0).

Why is this a problem? Again, when Router A sends an IGRP update to Router B via Serial, it will not include the subnet mask for the 10.1.0.0 network. Instead, Router A will consider itself a **border** router, as the 10.1.0.0 and 192.168.123.0 networks *do not* belong to the same major network. Router A will **summarize** the 10.1.0.0/16 network to its classful boundary of 10.0.0.0/8.

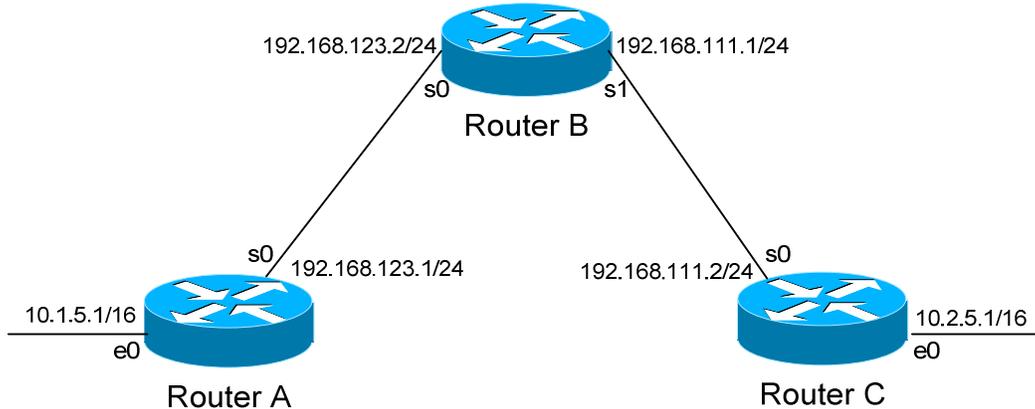
Router B will accept this routing update, and realize that it does not have a directly connected interface in the 10.x.x.x scheme. Thus, it has no subnet mask to apply to this route. Because of this, Router B will install the summarized 10.0.0.0 route into its routing table.

Router C, similarly, will consider itself a border router between networks 10.2.0.0 and 192.168.111.0. Thus, Router C will *also* send a summarized 10.0.0.0 route to Router B.

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto: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>.

**Limitations of IGRP (continued)**

Router B's routing table will then look like:

```
RouterB# show ip route
```

```
Gateway of last resort is not set
```

```
C    192.168.123.0 is directly connected, Serial0
C    192.168.111.0 is directly connected, Serial1
I    10.0.0.0 [120/1] via 192.168.123.1, 00:00:00, Serial0
      [120/1] via 192.168.111.2, 00:00:00, Serial1
```

That's right, Router B now has two *equal* metric routes to get to the summarized 10.0.0.0 network, one through Router A and the other through Router C. Router B will now *load balance* all traffic to *any* 10.x.x.x network between routers A and C. Suffice to say, this is not a good thing. ☺

It gets better. Router B then tries to send routing updates to Router A and Router C, including the summary route of 10.0.0.0/8. Router A's routing table looks like:

```
RouterA# show ip route
```

```
Gateway of last resort is not set
```

```
C    192.168.123.0 is directly connected, Serial0
      10.0.0.0/16 is subnetted, 1 subnet
C    10.1.0.0 is directly connected, Ethernet0
```

Router A will receive the summarized 10.0.0.0/8 route from Router B, and will reject it. This is because it already has the summary network of 10.0.0.0 in its routing table, and it's directly connected. Router C will respond exactly the same, and the 10.1.0.0/16 and 10.2.0.0/16 networks will *never* be able to communicate.

\*\*\*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto: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>.

## IGRP Timers

IGRP has four basic timers:

**Update Timer** (default **90 seconds**) – indicates how often the router will send out a routing table update.

**Invalid Timer** (default **270 seconds**) – indicates how long a route will remain in a routing table before being marked as invalid, if no new updates are heard about this route. The invalid timer will be reset if an update is received for that particular route *before* the timer expires.

A route marked as invalid is *not* immediately removed from the routing table. Instead, the route is marked (and advertised) with a metric of 101 (remember, 100 maximum hops is default), indicating it is unreachable, and placed in a **hold-down** state.

**Hold-down Timer** (default **280 seconds**) – indicates how long IGRP will “suppress” a route that it has placed in a **hold-down** state. IGRP will not accept any new updates for routes in a hold-down state, until the hold-down timer expires.

A route will enter a hold-down state for one of three reasons:

- The invalid timer has expired.
- An update has been received from another router, marking that route with a metric of 101 (unreachable).
- An update has been received from another router, marking that route with a *higher* metric than what is currently in the routing table (this is to prevent loops).

**Flush Timer** (default **630 seconds**) – indicates how long a route can remain in a routing table before being flushed, if no new updates are heard about this route. The flush timer runs **concurrently with the invalid timer**, and thus will flush out a route 360 seconds after it has been marked invalid.

IGRP timers must be identical on **all** routers on the IGRP network, otherwise massive instability will occur.

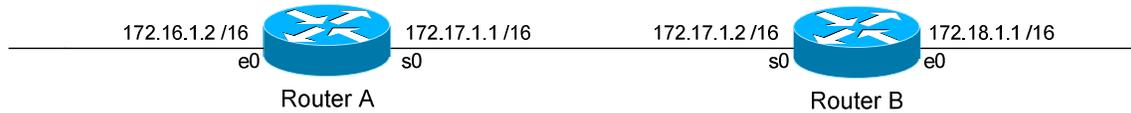
\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto: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>.

## IGRP Loop Avoidance Mechanisms

IGRP, as a Distance Vector routing protocol, is susceptible to loops.



Let's assume no loop avoidance mechanisms are configured on either router. If the 172.18.0.0 network fails, Router B will send out an update to Router A within 30 seconds (whenever its update timer expires) stating that route is unreachable.

But what if an update from Router A reaches Router B *before* this can happen? Router A believes it can reach the 172.18.0.0 network in one hop (through Router B). This will cause Router B to believe it can reach the failed 172.18.0.0 network in **two hops**, through Router A. Both routers will continue to increment the metric for the network until they reach an infinity hop count (by default, 101). This behavior is known as **counting to infinity**.

How can we prevent this from happening? There are several loop avoidance mechanisms:

**Split-Horizon** – Prevents a routing update from being sent out the interface it was received on. In our above example, this would prevent Router A from sending an update for the 172.18.0.0 network *back* to Router B, as it originally learned the route from Router B. Split-horizon is **enabled** by default on Cisco Routers.

**Route-Poisoning** – Works in conjunction with split-horizon, by **triggering** an automatic update for the failed network, without waiting for the update timer to expire. This update is sent out all interfaces with an infinity metric for that network.

**Hold-Down Timers** – Prevents IGRP from accepting any new updates for routes in a hold-down state, until the hold-down timer expires. If Router A sends an update to Router B with a *higher* metric than what is currently in Router B's routing table, that route will be placed in a hold-down state.

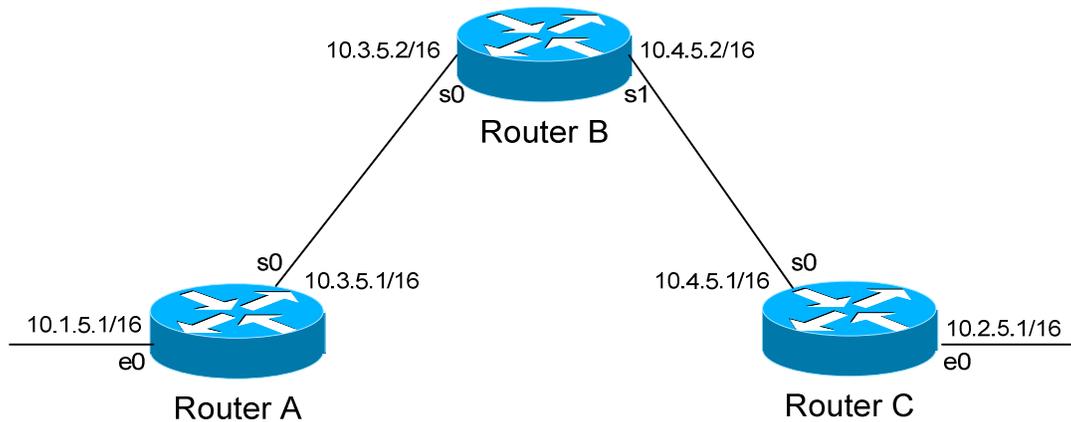
\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto: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>.

## IGRP Passive Interfaces

It is possible to control which router interfaces will participate in the IGRP process.



Consider the following scenario. Router C does not want to participate in the IGRP domain. However, it still wants to *listen* to updates being sent from Router B, just not *send* any updates back to Router B:

```

RouterC(config)# router igrp 10
RouterC(config-router)# network 10.4.0.0
RouterC(config-router)# network 10.2.0.0
RouterC(config-router)# passive-interface s0
  
```

The *passive-interface* command will prevent updates from being **sent** out of the Serial0 interface, but Router C will still **receive** updates on this interface.

We can configure **all** interfaces to be passive using the *passive-interface default* command, and then individually use the *no passive-interface* command on the interfaces we **do** want updates to be sent out:

```

RouterC(config)# router igrp 10
RouterC(config-router)# network 10.4.0.0
RouterC(config-router)# network 10.2.0.0
RouterC(config-router)# passive-interface default
RouterC(config-router)# no passive-interface e0
  
```

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto: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>.

## Advanced IGRP Configuration

To change the maximum hop-count to 255 for IGRP:

```
Router(config)# router igrp 10  
Router(config-router)# metric maximum-hops 255
```

\* \* \*

All original material copyright © 2012 by Aaron Balchunas ([aaron@routeralley.com](mailto: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>.