Default & Static Routes and Routing Information Protocol

Presented by: Mohammed Hamad
Introduction

• When a device has multiple paths to reach a destination, it always selects one path by preferring it over others. This selection process is termed as Routing.

• Routing is done by special network devices called routers, a router is always configured with some default route.

• A default route tells the router where to forward a packet if there is no route found for specific destination, in case there are multiple path existing to reach the same destination, router can make decision based on the metrics used in IP routing protocols:
  ■ **Hop count** - A simple metric that counts the number of routers a packet must traverse
  ■ **Bandwidth** - Influences path selection by preferring the path with the highest bandwidth
  ■ **Load** - Considers the traffic utilization of a certain link
  ■ **Delay** - Considers the time a packet takes to traverse a path
  ■ **Reliability** - Assesses the probability of a link failure, calculated from the interface error count or previous link failures
  ■ **Cost** - A value determined either by the IOS or by the network administrator to indicate preference for a route. Cost can represent a metric, a combination of metrics or a policy.

• Routes can be statically configured or dynamically learnt. One route can be configured to be preferred over others.
Routing Types:

Unicast routing

- Most of the traffic on the internet and intranets known as unicast data or unicast traffic is sent with specified destination.
- Routing unicast data over the internet is called unicast routing, hence the router just has to look up the routing table and forward the packet to next hop.

Broadcast routing

- By default, the broadcast packets are not routed and forwarded by the routers on any network.
- Reverse path forwarding is a technique, in which router knows in advance about its predecessor from where it should receive broadcast. This technique is used to detect and discard duplicates.
Multicast Routing

- Multicast routing is a special case of broadcast routing. In multicast routing, the data is sent to only nodes that want to receive the packets. The router must know that there are nodes, then only it should forward. Multicast routing works using spanning tree protocols to avoid looping, and multicast routing also uses reverse path forwarding techniques to detect and discard duplicates and loops.

Anycast Routing

- Anycast packet forwarding is a mechanism where multiple hosts can have the same logical address. When a packet destined to this logical address is received, it is sent to the host which is nearest in routing topology.
- Anycast routing is done with the help of DNS servers. Whenever an Anycast packet is received, it is enquired with DNS to find where to send it. DNS provides the IP address which is the nearest configured IP.
Introduction Cont.

Unicast Routing Protocols

There are two kinds of routing protocols to route unicast packets:

Distance Vector Routing Protocol

- Distance Vector is simple routing protocol which takes routing decision based on the number of hops between source and destination.
- A route with less number of hops is considered as the best route, for example Routing Information Protocol RIP.

Link State Routing Protocol

- Link State protocol is slightly complicated protocol than Distance Vector, it takes into account the states of links of all the routers in a network, all routers then calculate their best path for routing purposes, for example Open Shortest Path First OSPF and Intermediate System to Intermediate System ISIS.
**Static Routing**

Why Use Static Routing?

Static routing provides some advantages over dynamic routing, including:

- Static routes are not advertised over the network, resulting in better security.
- Static routes use less bandwidth than dynamic routing protocols, no CPU cycles are used to calculate and communicate routes.
- The path of the static route which uses to send data is known.
Static Routing Cont.

When to Use Static Routes:

Static routing has three primary uses:

- Providing ease of routing table maintenance in smaller networks that are not expected to grow significantly.
- Routing to and from stub networks. A stub network is a network accessed by a single route, and the router has only one neighbor.
- Using a single default route to represent a path to any network that does not have a more specific match with another route in the routing table. Default routes are used to send traffic to any destination beyond the next upstream router.
Types of Static Routes:

■ **Standard static route**
  Both IPv4 and IPv6 support the configuration of static routes. Static routes are useful when connecting to a specific remote network.

■ **Default static route**
  A default static route is a route that matches all packets. A default route identifies the gateway IP address to which the router sends all IP packets that it does not have a learned or static route. A default static route is simply a static route with 0.0.0.0/0 as the destination IPv4 address. Configuring a default static route creates a Gateway of Last Resort.
Static Routing Cont.

■ Summary static route

To reduce the number of routing table entries, multiple static routes can be summarized into a single static route if:

➢ The destination networks are contiguous and can be summarized into a single network address.
➢ The multiple static routes all use the same exit interface or next-hop IP address.
Floating static route

- Floating static routes are static routes that are used to provide a backup path to a primary static or dynamic route, in the event of a link failure. The floating static route is only used when the primary route is not available.
- To accomplish this, the floating static route is configured with a higher administrative distance than the primary route. Recall that the administrative distance represents the trustworthiness of a route.
- If multiple paths to the destination exist, the router will choose the path with the lowest administrative distance.
<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to implement in a small network.</td>
<td>Suitable for simple topologies or for special purposes such as a default static route.</td>
</tr>
<tr>
<td>Very secure. No advertisements are sent, unlike with dynamic routing protocols.</td>
<td>Configuration complexity increases dramatically as the network grows. Managing the static configurations in large networks can become time consuming.</td>
</tr>
<tr>
<td>It is very predictable, as the route to the destination is always the same.</td>
<td>If a link fails, a static route cannot reroute traffic. Therefore, manual intervention is required to re-route traffic.</td>
</tr>
<tr>
<td>No routing algorithm or update mechanisms are required. Therefore, extra resources (CPU and memory) are not required.</td>
<td></td>
</tr>
</tbody>
</table>
Configuring a Static Routes:

There are two ways of configuring a static route.

- The first uses the next hop router's IP address on the connected network:
  
  \[ R1(config)#ip\ route <destination network> <subnet mask> <next hop router address> \]
  
  \[ R1(config)#ip\ route 192.168.2.0 255.255.255.0 192.168.51.2 \]

- The second uses the router's own exit interface. This way is faster for the router because it doesn't have to first look up the exit interface from the connected network:
  
  \[ R1(config)#ip\ route <destination network> <subnet mask> <local router exit interface> \]
  
  \[ R1(config)#ip\ route 192.168.2.0 255.255.255.0 fa1/0 \]
Static Routing Cont.

R1(config)#ip route 192.168.2.0 255.255.255.0 192.168.51.2
R1(config)#ip route 192.168.2.0 255.255.255.0 fa2/0

**R1’s exit interface is fa2/0**

The next hop router interface towards the 192.168.2.0 network is 192.168.51.2
Configure IPv6 Static Routes

- Static routes for IPv6 are configured using the `ipv6 route` global configuration command.
- `Router(config)# ipv6 route ipv6-prefix/prefix-length { ipv6-address | exit-intf }
- IPv6 static routes can also be implemented as:
  - Standard IPv6 static route
  - Default IPv6 static route
  - Summary IPv6 static route
  - Floating IPv6 static route
- As with IPv4, these routes can be configured as recursive, directly connected, or fully specified.
- The `ipv6 unicast-routing` global configuration command must be configured to enable the router to forward IPv6 packets.
Defaults Routing

- Normally, if a specific route to a particular network does not exist, a router will drop all traffic destined to that network. A **default route**, or **gateway of last resort**, allows traffic to be forwarded, even without a specific route to a particular network.

- The default route is identified by all zeros in both the network and subnet mask (\(0.0.0.0\ 0.0.0.0\)). It is the *least* specific route possible, and thus will only be used if a more specific route does not exist (hence “gateway of last resort”).
Configuring Defaults Routes

- In the diagram below R1 needs a default route or gateway of last resort configured so that it can route traffic to unknown networks across the internet. If R1 does not have a default route, traffic to all unknown networks will be dropped, and surfing the Web will not be possible. To configure a default route you must configure a static route to the 0.0.0.0 network and 0.0.0.0 subnet mask to the next hop router or exit interface which has a path out of the network (see below).

- The commands to create a default route or gateway of last resort are:

  \[R1(config)#ip route 0.0.0.0 0.0.0.0 \textless\text{next hop router IP address}\]  
  \[R1(config)#ip route 0.0.0.0 0.0.0.0 192.168.50.2\]

  or

  \[R1(config)#ip route 0.0.0.0 0.0.0.0 \textless\text{exit interface}\]  
  \[R1(config)#ip route 0.0.0.0 0.0.0.0 \text{s2/0}\]
To reach the Internet R1 needs a default route.

```
ip route 0.0.0.0 0.0.0.0 s2/0
```

or

```
ip route 0.0.0.0 0.0.0.0 192.168.50.2
```
RIP (Routing Information Protocol)

RIP is a standardized Distance Vector protocol, designed for use on smaller networks. RIP was one of the first true Distance Vector routing protocols, and is supported on a wide variety of systems.

• RIP adheres to the following Distance Vector characteristics:
  • RIP sends out periodic routing updates (every 30 seconds)
  • RIP sends out the full routing table every periodic update
  • RIP uses a form of distance as its metric (in this case, hopcount)
  • RIP uses the Bellman-Ford Distance Vector algorithm to determine the best “path” to a particular destination.
RIP Cont.

- Other characteristics of RIP include:
  - RIP supports IP and IPX routing.
  - RIP utilizes UDP port 520
  - RIP routes have an administrative distance of 120.
  - RIP has a maximum hopcount of 15 hops.

- Any network that is 16 hops away or more is considered unreachable to RIP, thus the maximum diameter of the network is 15 hops. A metric of 16 hops in RIP is considered a poison route or infinity metric.
- If multiple paths exist to a particular destination, RIP will load balance between those paths (by default, up to 4) only if the metric (hopcount) is equal.
- RIP uses a round-robin system of load-balancing between equal metric routes.
There are several loop avoidance mechanisms:

**Split-Horizon**
- Prevents a routing update from being sent out the interface it was received on.
- 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 RIP 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.

**RIP Versions**

RIP has three versions, **Version 1 (RIPv1)** & **Version 2 (RIPv2)** and **RIPng** (RIP next generation).
RIP Version 1 (RIPv1)

RIPv1 Characteristics:

1. **Classful routing** - RIP is a classful routing protocol, it does not send or receive subnet masks, it assumes classful subnet masks.
2. **Periodic updates** - RIP broadcasts routing updates every 30 seconds. It broadcasts the entire routing table in the update.
3. **Triggered updates** - RIP also broadcasts updates when there is a change in the topology, like a network going up or down.
4. **Metric** - RIP's metric is hop count. A hop is a router, so a RIP route with metric of 2 means the network is two routers away. RIP's maximum distance is 15 hops. 16 hops is infinity and is used to mark a route as dead. A router that receives a packet then removed from the routing table.
5. **Transport Layer** - RIP sends out updates on UDP port 520.
6. **Administrative Distance** - RIP's administrative distance is 120. AD ranks the trustworthiness, or reliability of the route, the lower the administrative distance the better the route.

7. **Hold Down Timer** - RIP uses a hold down timer of 180 seconds so that it does not propagate bad routes and does not have a count-to-infinity routing loop. Route is flushed at 240 seconds.

8. **Load Balancing** - Default load balancing across 4 equal cost routes. Can go up to 6 routes.

9. **Count to infinity** - is prevented by hold down timers and by RIPs maximum metric of 15 hops.

10. **Routing Loops** - are prevented by: Hold Down Timer, Split Horizon Rule, Route Poisoning, Route Poisoning with Poison Reverse, as well as TTL.

11. **Automatic Route Summarization** - RIP automatically summarizes routes into classful network ranges because it will not allow non-classful network configuration or the propagation of non-classful subnet masks. It can only advertise classful networks.
IOS CLI commands to use with RIP

```
router(config)#router rip //to activate RIP
router(config-router)#network <network ip address> //to add a participating network and interface
router(config-router)#passive-interface <interface type> <interface number> //to stop RIP from sending updates out of an interface
router#show ip route //to verify your routing table
router#show ip protocols //to verify your RIP configuration
```
RIP Version 2 (RIPv2)

- RIPv2 has one great improvement over RIPv1, instead of sending just the network address in its routing updates it can sends the subnet mask and the next hop address as well, this means that it can advertise non-classful subnets. In this way, RIPv2 supports VLSM and CIDR.

- An important point to remember is the difference in sending and receiving version updates. RIPv1 sends version 1 updates, receives both version 1 and 2 updates, but will ignore the version 2 updates. RIPv2 sends and receives only version 2 updates.
Here is a list of the similarities and differences of RIPv2 versus RIPv1:

**RIPv2 Differences from RIPv1**

- **VLSM and CIDR** - is supported by sending the subnet mask and the next hop address in its routing updates.
- **Multicasts** - its routing updates to 224.0.0.9. instead of broadcasting to 255.255.255.255 like RIPv1
- **Authentication** - RIPv2 supports md5 authentication
- **Updates** - RIPv2 sends and receives version 2 updates only. RIPv1 sends version 1 updates and receives both 1 and 2, however version 2 information is ignored.

**RIPv2 Similarities to RIPv1**

- **Auto Summarizes** by default (You will need to turn this off if you have discontiguous networks)
- **Distance Vector Protocol**
- **Hop Count** is the metric with a maximum of 15 hops, 16 is infinity (dropped).
RIPv2 Cont.

- Cisco IOS CLI Commands to config RIP Protocol

  router(config)#router rip
  router(config-router)#version 2
  router(config-router)#no auto-summary
  router(config-router)#network <network address>
  router(config-router)#network <network address>
  router(config-router)#end
RIPv2 Cont.

**RIPv2 Authentication**

RIPv2 supports authentication to secure routing updates.

- The first step is creating a shared authentication *key* that must be identical on both routers. This is accomplished in global configuration mode:

  ```
  RouterA(config)# key chain MYCHAIN
  RouterA(config-keychain)# key 1
  RouterA(config-keychain-key)# key-string MYPASSWORD
  RouterB(config)# key chain MYCHAIN
  RouterB(config-keychain)# key 1
  RouterB(config-keychain-key)# key-string MYPASSWORD
  ```

  The first command creates a *key chain* called *MYCHAIN*. We must then associate a *key* to our keychain. Then we actually configure the shared key using the *key-string* command.
- We then apply our key chain to the interface connecting to the other router:

  **RouterA(config)# interface s0**
  **RouterA(config-if)# ip rip authentication key-chain MYCHAIN**
  **RouterB(config)# interface s0**
  **RouterB(config-if)# ip rip authentication key-chain MYCHAIN**

If there was another router off of Router B’s Ethernet port, we could create a separate key chain with a different key-string. Every router on the RIP domain does not need to use the same key chain, only interfaces directly connecting two (or more) routers.
The final step in configuring authentication is identifying which encryption to use. By default, the key is sent in clear text:

```
RouterA(config)# interface s0
RouterA(config-if)# ip rip authentication mode text
```

Or we can use MD5 encryption for additional security:

```
RouterA(config)# interface s0
RouterA(config-if)# ip rip authentication mode md5
```

Whether text or MD5 is used, it **must** be the same on both routers.
RIPng

- RIPng (RIP next generation), defined in [RFC 2080](https://tools.ietf.org/html/rfc2080), is an extension of RIPv2 for support of IPv6, the next generation Internet Protocol.
- RIPng sends updates on UDP port 521 using the multicast group FF02::9.

**The main differences between RIPv2 and RIPng are:**
- RIPv2 does not support IPv6 networking while RIPng supports it.
- RIPv2 supports RIPv1 updates authentication, while RIPng does not.
- RIPv2 encodes the next-hop into each route entry, while RIPng requires specific encoding of the next hop for a set of route entries.
RIPng Cont.

RIPng Configuration

The Basic RIPng Configu commands required to configure RIPng on a network device:

✓ Enable ipv6 Routing on all Routers:

Router # configure terminal (Enter global configuration mode )
Router (config) # ipv6 unicast-routing (Enable device IPv6 unicast forwarding )

✓ Configure and Enabling RIPng on all Interfaces of Routers:

Router (config) # ipv6 router rip process-name (Enter IPv6 RIP router configuration mode )
Router (config) # interface interface (Enter interface configuration mode )
Router (config-if) # ipv6 rip process-name enable (Enable RIPng on the interface )