



Global Knowledge®

Expert Reference Series of White Papers

Routing and Routing Protocols

Routing and Routing Protocols

Al Friebe, Global Knowledge Instructor, CCDA, CCDP, CCNA, CCNP, A+, CCSI

Introduction

In this white paper, we'll discuss routing and routing protocols. Before we begin, let's define what we mean by a route. In common usage, a route is an entry in the IP routing table.

Routing

You can display the IP routing table (available routes) with the command **show ip route**. Each entry in the routing table gives the best way for that router to reach a particular IP prefix. Remember that a prefix is a particular address/mask combination. Examples of prefixes are:

- 10.0.0.0/8 (a classful network)
- 172.168.100/24 (a subnet)
- 192.168.1.32/29 (another subnet)
- 200.100.200.3/32 (a host route)
- 0.0.0.0/0 (the default route)

For each prefix displayed in the routing table, the entry will indicate how the route was learned, the next hop router's address and/or outbound interface used to reach it, and other information. Although a routing table entry always represents the best known way to reach a particular prefix, the router may be aware of other possible paths to that prefix. If so, those additional paths would be tracked in other behind-the-scenes data structures separate from the routing table.

Once all routers have learned their best path(s) to all available prefixes, the network is "converged". After it's converged, the routers do not necessarily have identical routing tables, but they are consistent and correct.

When a router receives an update or change, it checks its routing table to see if the update contains an advertised prefix that was previously unknown. If so, that prefix is added to the routing table, with the advertising router as the best next hop. The time lag between the change and re-convergence is referred to as the convergence interval and is a function of the routing protocol(s) and the size of the network.

If a router receives an advertisement for a known prefix, the router checks to see if the advertised route has a better metric than the current route. If so, the router updates the routing table to use the advertising router as the next hop for that prefix. If not, the router ignores the advertisement.

When all change information has been passed around, and the routing tables have stabilized, the network is again converged. So that all routers become aware of changes to the topology in a timely fashion, the routing tables are advertised periodically.

Another View

Think of the internet network as being like a shopping mall: There's a map near each door that shows where all the stores are; each is identical except for the "You Are Here" marker. We can use the nearest map to find the best route from where we are to each of the stores that we'd like to visit.

Similarly, each router has a topology database. Because all of the routers are part of one internet network, all of the topology databases are identical. Using its copy of the topology database, each router individually calculates the best next hop for each known destination prefix, and places this information in its routing table. At this point, routing has converged.

There are a few differences between routing tables and our shopping mall analogy. First, the maps in the mall are diagrams, while routers keep the topology databases in table form. Second, while most shoppers are interested in visiting only a few stores within the mall (there may be exceptions), routers calculate the best next hop for all known prefixes.

There are three ways that a router can learn about the existence of a route.

- **Directly connected routes** are those prefixes to which the router has a direct physical connection. Assuming that the interface is "up/up", the router will calculate the prefix based on the address and mask configured on the interface, and place a C (Connected) route for that prefix in the routing table.
- **Static routes** are those that are configured by an administrator with the ip route command, instructing the router to use a particular next hop or outbound interface to reach a particular prefix. Assuming that the interface in question is "up/up", the router will place an S (Static) route for that prefix in the routing table.
- **Dynamic routes** are those learned via a routing protocol. The mechanism by which the router learns the route varies by routing protocol, as does the letter representing the way the route was learned. Examples include "R" (RIP), "O" (OSPF) and "D" (EIGRP) routes.

We can classify the dynamic route protocols several ways, but Distance-Vector (D-V) is the most common. (Other methods include Link-State, Hybrid, and Path-Vector.)

D-V protocols get their name from the fact that each update sent from router to router is a mathematical vector (a multi-valued variable) containing prefix and metric (distance) information. The metric varies by routing proto-

col, and includes such things as hop count, cost, bandwidth, and delay. The vectors used with routing protocols are mathematical vectors, not navigational vectors (such as Northeast).

D-V protocols work, but they're not perfect. Let's imagine that we have two routers (R1 and R2) and three subnets (10.1.0.0/16, 10.2.0.0/16, and 10.3.0.0/16), connected in a string.

- 10.1.0.0/16 – R1 – 10.2.0.0/16 – R2 – 10.3.0.0/16

Once the network has converged, both routers will know the best path to each of the three subnets. Looking specifically at the 10.3.0.0/16 subnet

- R2 sees 10.3.0.0/16 as directly connected (zero hops)
- R1 sees 10.3.0.0/16 as reachable via R2 (one hop)

Imagine now that the 10.3.0.0/16 subnet becomes unreachable, and R2 removes it from its routing table. Since we're running RIP, each router is periodically flooding its routing table. At this moment, R1, which is not yet aware of the change, sends its table to R2. As a result, R2, which can no longer reach 10.3.0.0/16 directly, now thinks that it can reach that subnet via R1, and that it is two hops away.

- R2 sees 10.3.0.0/16 as reachable via R1 (two hops)
- R1 sees 10.3.0.0/16 as reachable via R2 (one hop)

When R2 next floods its routing table, R1 will see that 10.3.0.0/16 is now reachable via R2, and that it's three hops away.

- R2 sees 10.3.0.0/16 as reachable via R1 (two hops)
- R1 sees 10.3.0.0/16 as reachable via R2 (three hops)

When R1 next floods its routing table, we'll have:

- R2 sees 10.3.0.0/16 as reachable via R1 (four hops)
- R1 sees 10.3.0.0/16 as reachable via R2 (three hops)

And after R2's next update:

- R2 sees 10.3.0.0/16 as reachable via R1 (four hops)
- R1 sees 10.3.0.0/16 as reachable via R2 (five hops)

There are two things to note about the current situation.

1. R1 thinks that it can reach 10.3.0.0/16 via R2, and R2 thinks that it can reach it via R1. The truth is that neither router can reach that subnet. The effect of this is that any packet destined for 10.3.0.0/16 will be caught in a routing loop between R1 and R2. This is bad because as R1 and R2 ping-pong the packet, it's wasting their time, as well as wasting bandwidth on the link between them. Fortunately, the packet will not loop forever; eventually the TTL of the packet will hit zero, at which time the packet will be discarded (see RFC 1812 for details).

2. The metrics are increasing with every update cycle. This behavior is referred to as the **count to infinity**, and it is a symptom of a routing loop in a D-V protocol. When the metric hits the maximum as determined by the number of bits in the metric field, it will next go back to zero, and the cycle will repeat.

It's important to note that for a particular data packet that's caught in the loop, the TTL field in the packet's IP header is counting **down** towards zero (at which point the packet is discarded), while the metric fields in the routing tables are counting **up** to the max (at which point the cycle repeats). While an individual packet does not loop forever, the loop does last forever.

One could make the case that since the 10.3.0.0/16 subnet is unreachable, the fact that traffic for that subnet loops is a non-issue. However, since the looping traffic gets in the way of other traffic, this is not the case. Thus, having routing loops for unreachable destinations is sub-optimal, and we want the entries for any unreachable subnets removed from all routing tables as quickly as possible.

Link-State Protocols

With the D-V protocols, as the size of an internetwork grows, the number of prefixes to be advertised and stored grows, requiring greater bandwidth and RAM. Also, as the number of routing updates increases, more CPU power is required to send and receive the updates.

Around 1990, based on the projected growth of internetworks, it became apparent that the scalability of the D-V protocols would present a problem. To deal with this, an entirely new category of routing protocols was developed: the Link-State protocols.

Each Link-State router builds a topology database that contains everything that router knows about the topology of the internetwork. This information includes which IP prefixes are directly connected to which routers and the metric information for each prefix. It's called Link-State protocol because each router knows the detailed prefix (link) and metric (state) information - in other words, the state of the links.

It's a little like a jigsaw puzzle, where each router contributes a piece of the puzzle (the prefixes to which it is directly connected). The goal is to collect all of the pieces, and then assemble the puzzle.

Each router begins by placing an entry for itself in its topology database. Next, the router uses a "hello" protocol to discover its directly connected neighbors. The neighbors then exchange topology information.

When an update containing new topology information is received, the router adds that piece to its topology database, and then floods that piece. Each neighbor does the same. When this process is complete, all routers will have detailed information about the entire topology, and each router can then determine the best path from itself to each destination prefix.

With a Link-State protocol, what's being advertised from one router to another is raw topology information, not routing table entries. As a result, each Link-State router knows the entire topology of the internetwork. Since

the topology updates are acknowledged (making the protocol reliable), there is no need for frequent periodic updates. Instead, updates need only be sent when a topology change occurs. In between changes, the hello protocol is used to verify the continued availability of neighbors.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge. Check out the following Global Knowledge course(s):

ROUTE – Implementing Cisco IP Routing v1.0

MPLS – Implementing Cisco MPLS v2.2

BCN – Building Core Networks with OSPF, IS-IS, BGP, and MPLS Boot Camp v6

For more information or to register, visit www.globalknowledge.com or call **1-800-COURSES** to speak with a sales representative.

Our courses and enhanced, hands-on labs and exercises offer practical skills and tips that you can immediately put to use. Our expert instructors draw upon their experiences to help you understand key concepts and how to apply them to your specific work situation. Choose from our more than 1,200 courses, delivered through Classrooms, e-Learning, and On-site sessions, to meet your IT and business training needs.

About the Author

Al Friebe (CCDA, CCDP, CCNA, CCNP, A+, CCSI) has taught networking classes since 1995. He previously served as Global Knowledge's Course Director for BGP and BSCI, and he is the author of our current ICND2 labs. His previous experience includes instructor duty in the U.S. Navy's Nuclear Power School, radio-chemistry, software engineering, and network management.