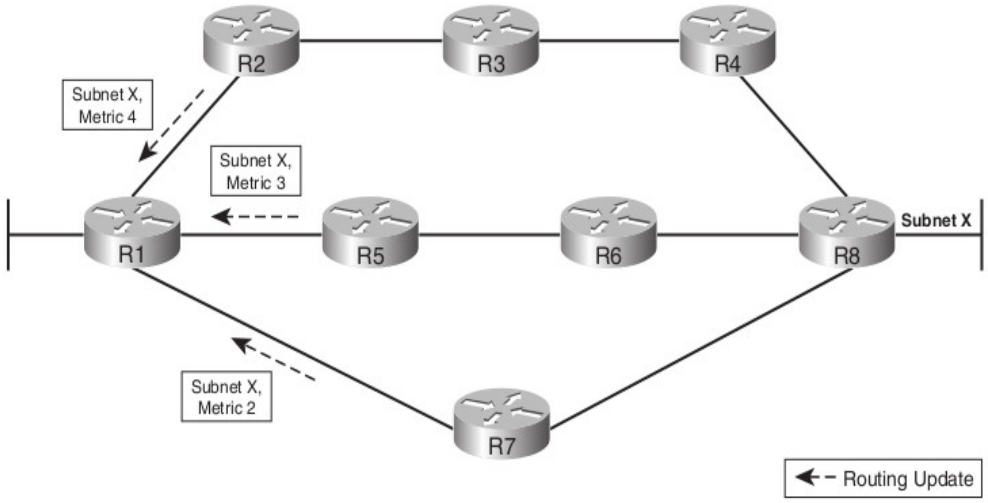
DISTANCE VECTOR ROUTING ALGORITHM

As the name implies, distance vector means that routes are advertised as vectors of distance and direction. Distance is defined in terms of a metric such as hop count and direction is simply the next-hop router or exit interface. A router using a distance vector routing protocol does not have the knowledge of the entire path to a destination network. Instead the router knows only:

The direction or interface in which packets should be forwarded and
The distance or how far it is to the destination network.

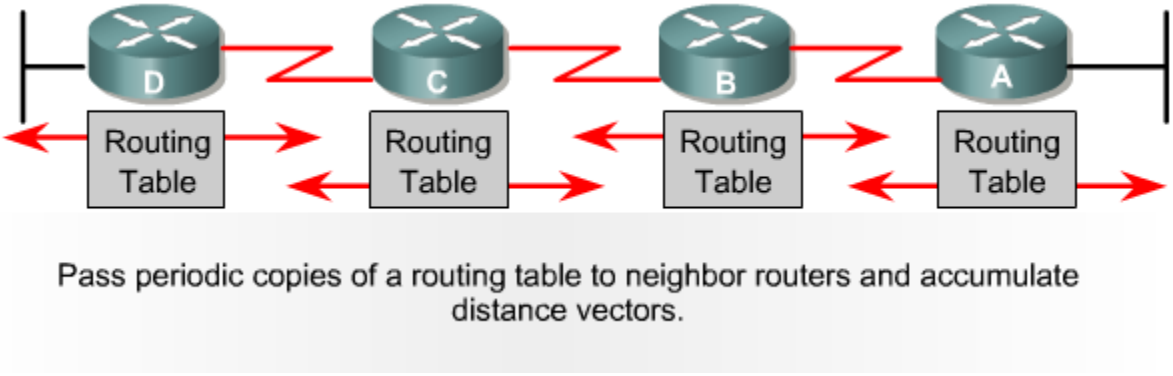
To show you more exactly what a distance vector protocol does, Figure shows a view of what a router learns with a distance vector routing protocol. The figure shows an internetwork in which R1 learns about three routes to reach subnet X:

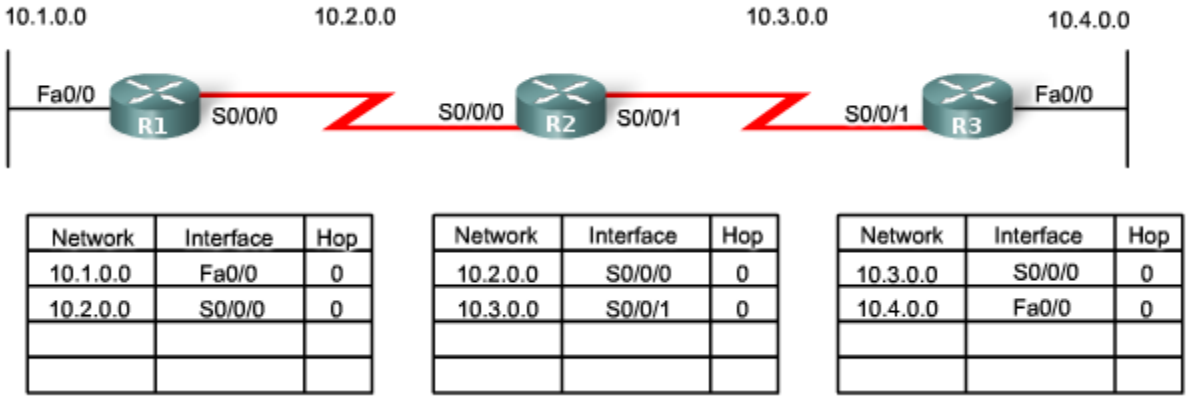
- The four-hop route through R2
- The three-hop route through R5
- The two-hop route through R7



R1 learns about the subnet, and a metric associated with that subnet, and nothing more. R1 must then pick the best route to reach subnet X. In this case, it picks the two-hop route through R7, because that route has the lowest metric.

Distance vector protocols typically use the **Bellman-Ford algorithm** for the best path route determination.

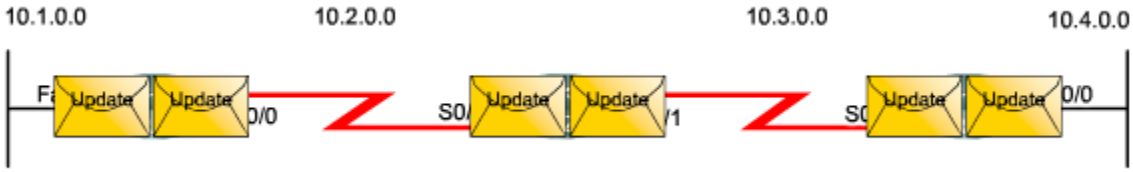




Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0

Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0

Network	Interface	Hop
10.3.0.0	S0/0/0	0
10.4.0.0	Fa0/0	0



Initial Update:

R1

- Sends an update about network 10.1.0.0 out the Serial0/0/0 interface
- Sends an update about network 10.2.0.0 out the FastEthernet0/0 interface
- Receives update from R2 about network 10.3.0.0 with a metric of 1
- Stores network 10.3.0.0 in the routing table with a metric of 1

R2

- Sends an update about network 10.3.0.0 out the Serial 0/0/0 interface
- Sends an update about network 10.2.0.0 out the Serial 0/0/1 interface
- Receives an update from R1 about network 10.1.0.0 with a metric of 1
- Stores network 10.1.0.0 in the routing table with a metric of 1
- Receives an update from R3 about network 10.4.0.0 with a metric of 1
- Stores network 10.4.0.0 in the routing table with a metric of 1

Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1

Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

Network	Interface	Hop
10.3.0.0	S0/0/0	0
10.4.0.0	Fa0/0	0
10.2.0.0	S0/0/1	1

R3

- Sends an update about network 10.4.0.0 out the Serial 0/0/0 interface
- Sends an update about network 10.3.0.0 out the FastEthernet0/0
- Receives an update from R2 about network 10.2.0.0 with a metric of 1
- Stores network 10.2.0.0 in the routing table with a metric of 1

After this first round of update exchanges, each router knows about the connected networks of their directly connected neighbors. However, did you notice that R1 does not yet know about 10.4.0.0 and that R3 does not yet know about 10.1.0.0? Full knowledge and a converged network will not take place until there is another exchange of routing information.

Next Update:
R1
Sends an update about network 10.1.0.0 out the Serial 0/0/0 interface.
Sends an update about networks 10.2.0.0 and 10.3.0.0 out the FastEthernet0/0 interface.
Receives an update from R2 about network 10.4.0.0 with a metric of 2.
Stores network 10.4.0.0 in the routing table with a metric of 2.
Same update from R2 contains information about network 10.3.0.0 with a metric of 1. There is no change; therefore, the routing information remains the same.

R2
Sends an update about networks 10.3.0.0 and 10.4.0.0 out of Serial 0/0/0 interface.
Sends an update about networks 10.1.0.0 and 10.2.0.0 out of Serial 0/0/1 interface.
Receives an update from R1 about network 10.1.0.0. There is no change; therefore, the routing information remains the same.
Receives an update from R3 about network 10.4.0.0. There is no change; therefore, the routing information remains the same.

Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	Fa0/0	0
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

R3

- Sends an update about network 10.4.0.0 out the Serial 0/0/0 interface.
- Sends an update about networks 10.2.0.0 and 10.3.0.0 out the FastEthernet0/0 interface.
- Receives an update from R2 about network 10.1.0.0 with a metric of 2.
- Stores network 10.1.0.0 in the routing table with a metric of 2.
- Same update from R2 contains information about network 10.2.0.0 with a metric of 1. There is no change; therefore, the routing information remains the same.

Note: Distance vector routing protocols typically implement a technique known as split horizon. Split horizon prevents information from being sent out the same interface from which it was received. For example, R2 would not send an update out Serial 0/0/0 containing the network 10.1.0.0 because R2 learned about that network through Serial 0/0/0.

-
-
-
-

Source :http://dayaramb.files.wordpress.com/2011/03/computer-network-notes-pu.pdf