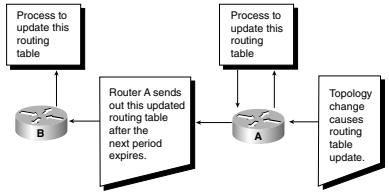
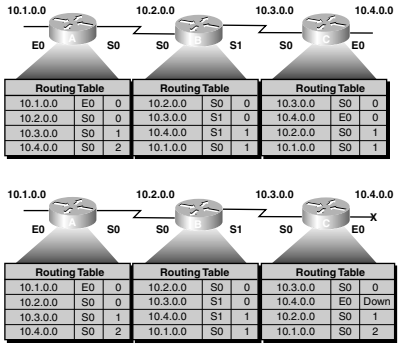
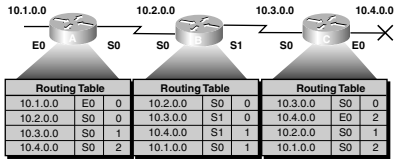
The process of updating routing tables is as follows: A router compares the information contained in the update to its current table. If the update contains information about a better route to some destination (a smaller total metric), the router updates its own routing table. As always, the distance metric is incremented 1 greater than the neighbor it received the update from. During updates, the router sends its entire routing table to each of its adjacent neighbors. The table includes the total path cost (defined by its metric) and the logical address of the first router on the path to each destination network. In the figure, Router B is one unit of cost from Router A. Therefore, it adds 1 to all costs reported by Router A.



How Routing Loops Occur in Distance Vector



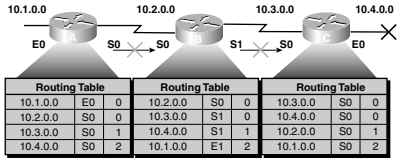
During updates, routing loops can occur if the network has inconsistent routing entries. Slow convergence on a new configuration is one cause of this phenomenon. The network is converged when all routers have consistent routing tables. The figure illustrates how a routing loop occurs. Before a network failure, all routers have correct tables. The figure uses hop count as a cost metric, so the cost of each link is 1. Router C is directly connected to network 10.4.0.0 with a distance of 0. Router A's path to network 10.4.0.0 is through Router B, with a hop count of 2.



If network 10.4.0.0 fails, Router C detects the failure and stops routing packets to that network. At this point, Routers A and B do not know of the failure. Router A's table still shows a valid path to 10.4.0.0 through Router B. If Router B sends out its normal update to A and C, Router C sees a valid path to 10.4.0.0 through Router B and updates its routing table to reflect a path to network 10.4.0.0 with a hop count of 2. (Remember, it must add 1 to the hop count.) Now router C sends an update to Router B, which then updates Router A. Router A detects the modified distance vector to network 10.4.0.0 and recalculates its own distance vector to 10.4.0.0 as 4. With each update, the incorrect information continues to bounce between the routers. Without some mechanism to prevent this, the updates continue. This condition, called *count to infinity,* continuously loops packets around the network.

Some distance vector routing algorithms have a Time-To-Live (TTL) value, but the routing loop problem can occur with this limit. You can avoid this by defining infinity as some maximum number. When you set this value, the routing loop continues until the metric exceeds the maximum allowed value. After the metric value exceeds the maximum, the network is unreachable.

Eliminating Routing Loops Through Split Horizon



Split horizon is one way to eliminate routing loops and speed up convergence. The idea behind split horizon is that it is never useful to send information about a route back in the direction from which the update came. If the router has no valid alternative path to the network, it is considered inaccessible. Split horizon also eliminates unnecessary routing updates, thus speeding convergence.

Route Poisoning

Route poisoning, which is part of split horizon, also eliminates routing loops caused by inconsistent updates. Route poisoning basically locks the table (using hold-down timers) until the network has converged.