Example: In the the Distance Vector algorithm, a node *x* updates its distance-vector estimate when it either sees a cost change in one of its directly attached links or receives a distance vector update from some neighbor. But to update its own forwarding table for a given destination *y*, what node *x* really needs to know is not the shortest- path distance to *y* but instead the neighboring node *v\**(*y*) that is the next-hop router along the shortest path to *y*.

Example: consider a network with simple three node

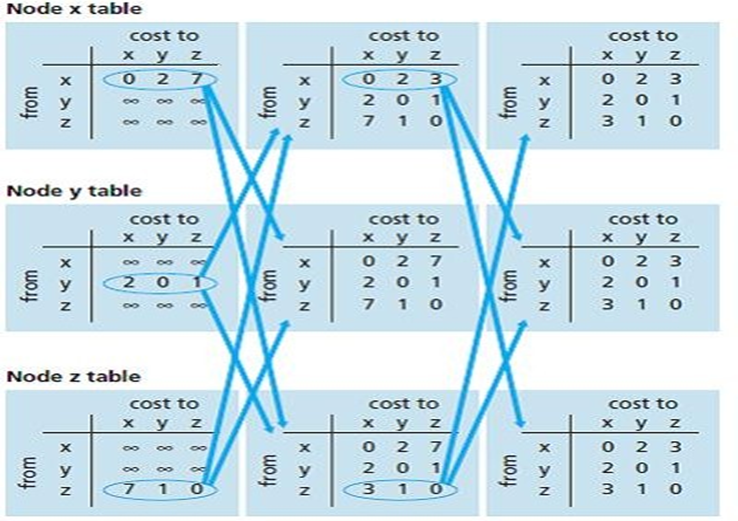
The operation of the algorithm is illustrated in a synchronous manner, where all nodes simultaneously receive distance vectors from their neighbors, compute their new distance vectors, and inform their neighbors if their distance vectors have changed.

The leftmost column of the figure displays three initial **routing tables** for each of the three nodes.

For example, the table in the upper-left corner is node *x*’s initial routing table. Within a specific routing table, each row is a distance vector—specifically, each node’s routing table includes its own distance vector and that of each of its neighbors.

Thus, the first row in node *x*’s initial routing table is ***D****x* = [*Dx*(*x*)*, Dx*(*y*)*, Dx*(*z*)] = [0, 2, 7].

Because at initialization node *x* has not received anything from node *y* or *z*, the entries in the second and third rows are initialized to infinity.



After initialization, each node sends its distance vector to each of its two neighbors. This is illustrated in above figure by the arrows from the first column of tables to the second column of tables. For example, node *x* sends its distance vector ***D****x* = [0, 2, 7] to both nodes *y* and *z*. After receiving the updates, each node recomputes its own distance vector. For example, node *x* computes

***Dx*(*x*) = 0**

***Dx*(*y*) *=* min{*c*(*x,y*) *+ Dy*(*y*), *c*(*x,z*) *+ Dz*(*y*)} = min{2 + 0, 7 + 1} = 2**

***Dx*(*z*) *=* min{*c*(*x,y*) *+ Dy*(*z*), *c*(*x,z*) *+ Dz*(*z*)} = min{2 + 1, 7 + 0} = 3**

The second column therefore displays, for each node, the node’s new distance vector along with distance vectors just received from its neighbors.

Note, for example,that node *x*’s estimate for the least cost to node *z*, *Dx*(*z*)*,* has changed from 7 to 3. After the nodes recompute their distance vectors, they again send their updated distance vectors to their neighbours (if there has been a change).