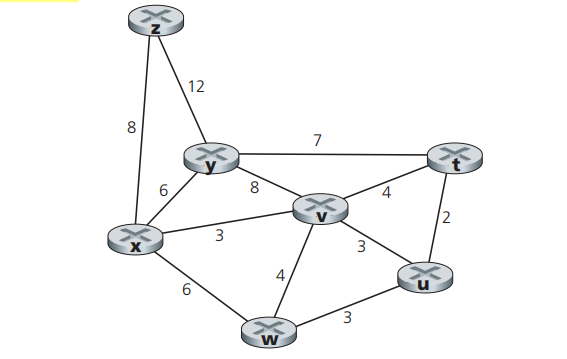
P3. Consider the following network. With the indicated link costs, use Dijkstra’sshortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to Table 5.1.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| step | N’ | D(z),p(z) | D(y),p(y) | D(v),p(v) | D(w),v(w) | D(u),v(u) | D(t),v(t) |
| 0 | x | 8,x | 6,x | 3,x | 6,x | ∞ | ∞ |
| 1 | x,v | 8,x | 6,x |  | 6,x | 6,v | 7,v |
| 2 | x,v,y | 8,x |  |  | 6,x | 6,v | 7,v |
| 3 | x,v,y,w | 8,x |  |  |  | 6,v | 7,v |
| 4 | x,v,y,w,u | 8,x |  |  |  |  | 7,v |
| 5 | X,v,y,w,u,t | 8,x |  |  |  |  |  |
| 6 | X,v,y,w,y,t,z |  |  |  |  |  |  |

So the following are the shortest paths from x along:

v:xv=3

y:xy=6

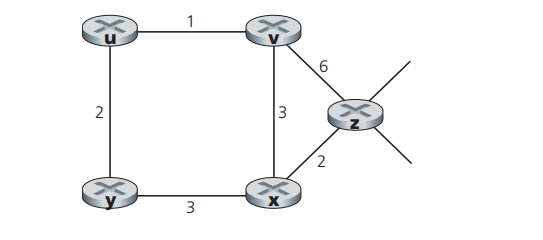
w:xw=6

u:xvu=6

t:xvt=7

z:xz=8

P5. Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node z.



For node z, the distance-vector table is shown below.

At the begging, z only know the distance to v and x.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | z | x | y | u | v |
| z | 0 | 2 | NA | NA | 6 |
| x | NA | NA | NA | NA | NA |
| v | NA | NA | NA | NA | NA |

After the first iteration, which is updated according to the neighboring nodes’ table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | z | x | y | u | v |
| z | 0 | 2 | 5 | 7 | 5 |
| x | 2 | 0 | 3 | NA | 3 |
| v | 6 | 3 | NA | 1 | 0 |

After the second iteration.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | z | x | y | u | v |
| z | 0 | 2 | 5 | 6 | 5 |
| x | 2 | 0 | 3 | 4 | 3 |
| v | 5 | 3 | 3 | 1 | 0 |

After the third iteration.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | z | x | y | u | v |
| z | 0 | 2 | 5 | 6 | 5 |
| x | 2 | 0 | 3 | 4 | 3 |
| v | 5 | 3 | 3 | 1 | 0 |

the distance table for all nodes are not changed in the third iteration, so the distance table entries at node z is the table above.

P9. Consider the count-to-infinity problem in the distance vector routing. Will the count-to-infinity problem occur if we decrease the cost of a link? Why? How about if we connect two nodes which do not have a link?

No, both will not lead to the count-to-infinity problem. If we decrease the cost of a link or connect two nodes which do not have a link, the vector table will be changed and inform its neighbor. After finite iterations, all nodes will find their new least costs and send their updates for distance vector.

The count-to-infinity problem occurs mainly due to increase of cost.