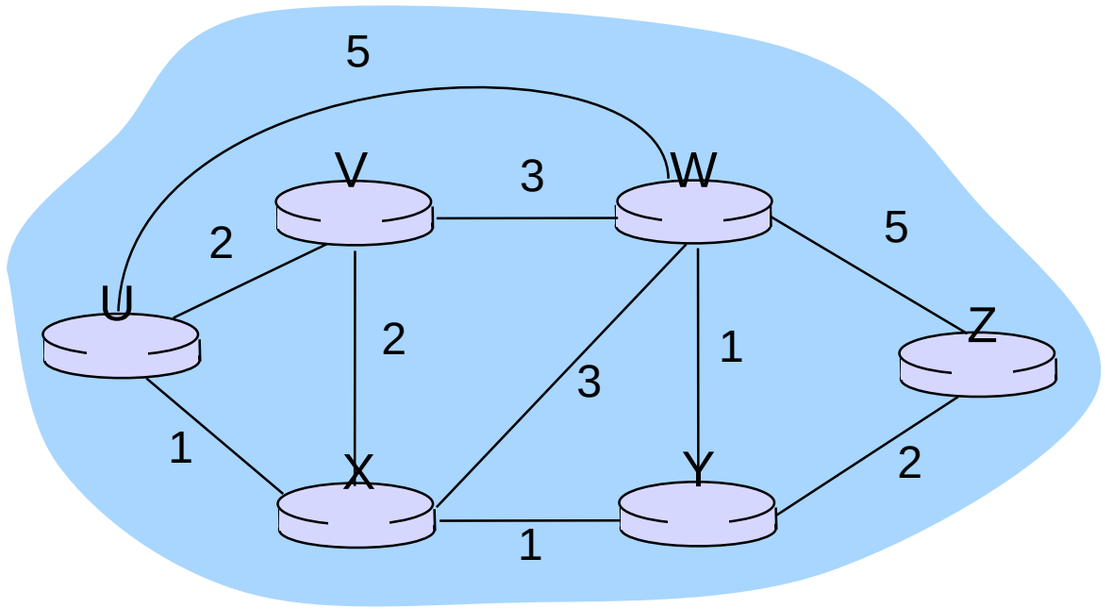
Dijkstra's Algorithm

- **Algorithm works as follows:**
  - Starts by assigning some initial distance value for each node in the graph
    - Distance from node s to itself is 0
    - Distances from node s to all other nodes in graph are initialized to INFINITY
  - Operates in steps, where at each step the algorithm improves the distance values for nodes in the graph
  - At each step the shortest distance from node s to another node in the graph is determined

Dijkstra's Algorithm Example

- **Network is represented as a graph**
  - Routers are nodes in graph
  - Links are edges in graph
    - Cost of edge is labeled



- **Each router computes distance to all other routers in network**
  - Example shows computation done by router U

A link-state routing algorithm

Dijkstra's algorithm

- **net topology, link costs known to all nodes**
  - accomplished via "link state broadcast"
  - all nodes have same info
- **computes least cost paths from one node ('source") to all other nodes**
  - gives *forwarding table* for that node
- **iterative: after k iterations, know least cost path to k dest.'s**

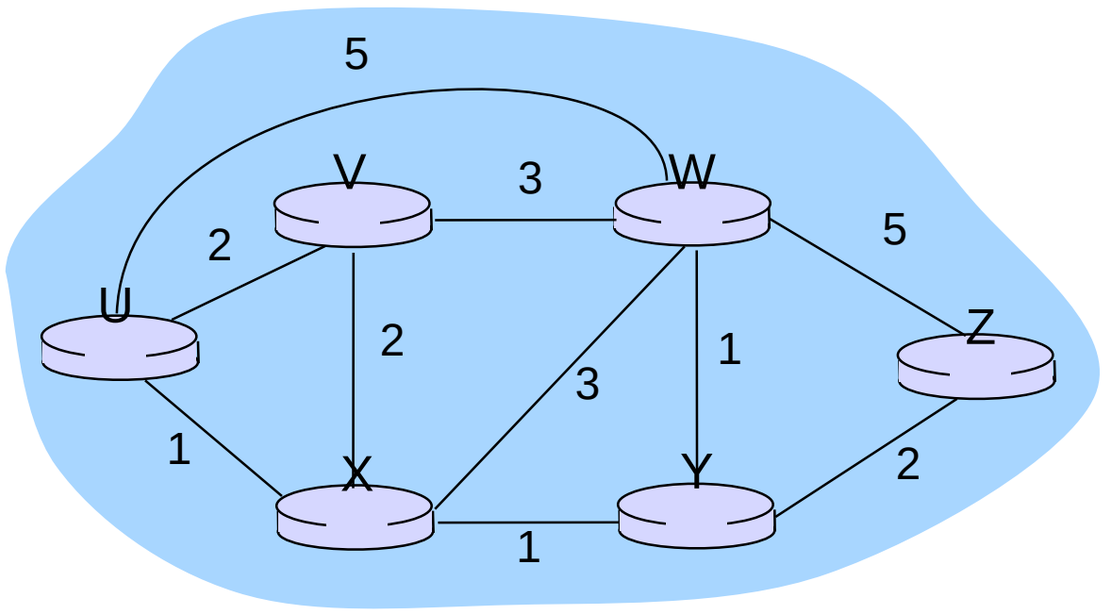
notation:

- **c(x,y): link cost from node x to y; = ∞ if not direct neighbors**
- **D(v): current value of cost of path from source to dest. v**
- **p(v): predecessor node along path from source to v**
- **N': set of nodes whose least cost path definitively known**

Dijsktra's algorithm

```
1  Initialization:
2    N' = {u}
3    for all nodes v
4      if v adjacent to u
5           then D(v) = c(u,v)
6      else D(v) = ∞
7
8   Loop
9     find w not in N' such that D(w) is a minimum
10    add w to N'
11    update D(v) for all v adjacent to w and not in N' :
12       D(v) = min( D(v), D(w) + c(w,v) )
13    /* new cost to v is either old cost to v or known
14     shortest path cost to w plus cost from w to v */
15  until all nodes in N'
```

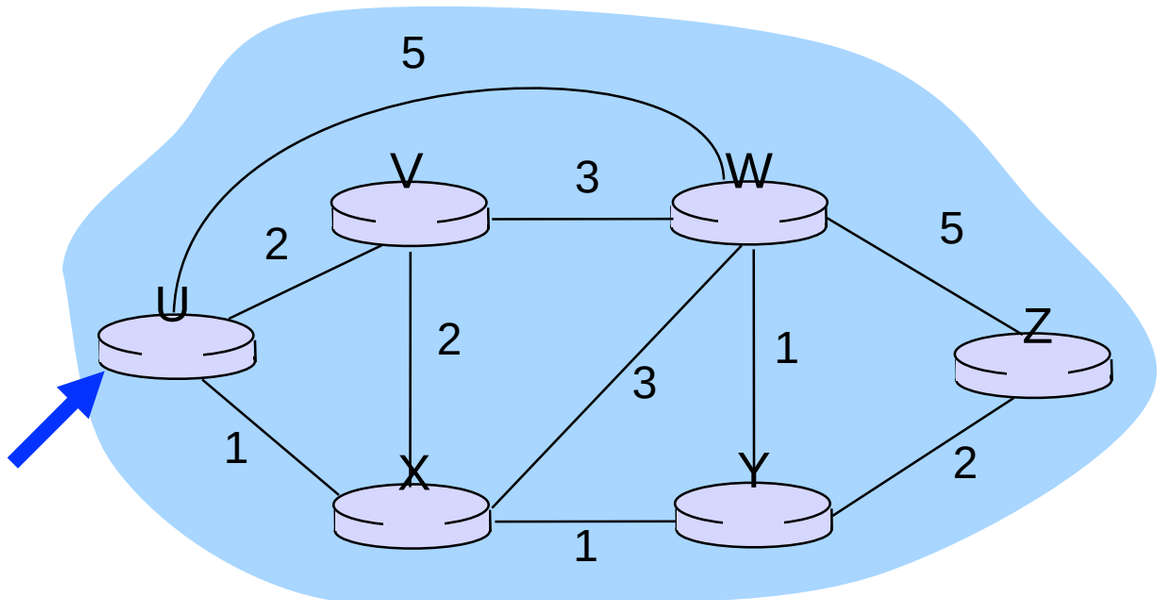
Dijkstra's Algorithm Example (Node U)



Initialize distances to U:
- Distance to itself is 0
- Distance to all other nodes is ∞

Node	Distance Computation from U						Path
U	0						
V	∞						
W	∞						
X	∞						
Y	∞						
Z	∞						

Dijkstra's Algorithm Example (Node U)

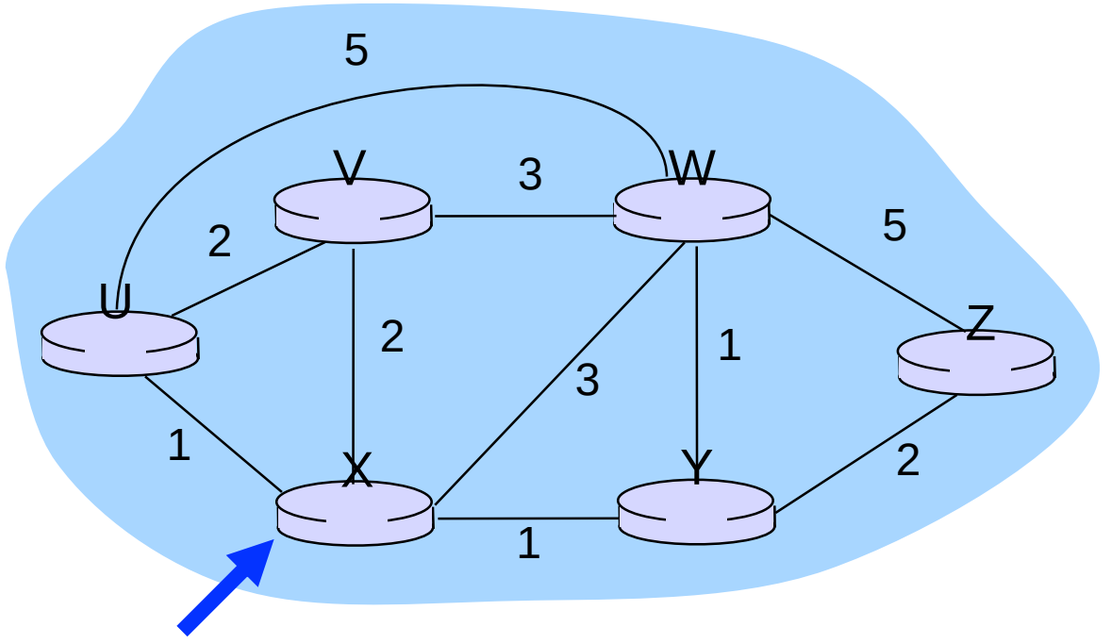


Select node with shortest distance to U (currently U) and determine shortest distance of its neighbors from U

- If node is unreachable it is still ∞
- Record the path

Node	Distance Computation from U						Path
U	0	-					-
V	∞	2					U ➔ V
W	∞	5					U ➔ W
X	∞	1					U ➔ X
Y	∞	∞					-
Z	∞	∞					-

Dijkstra's Algorithm Example (Node U)

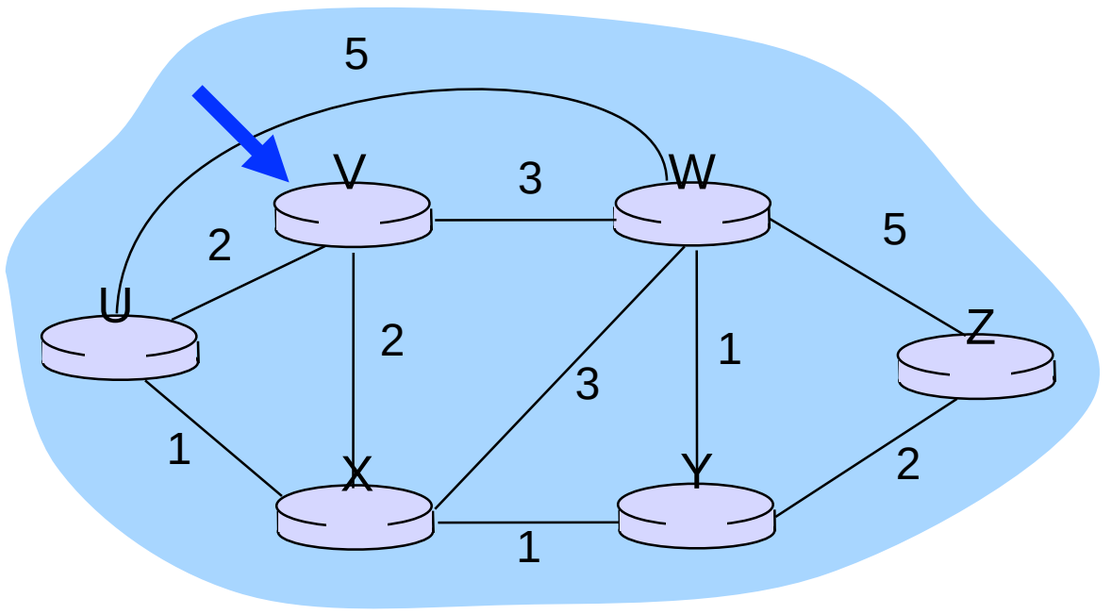


Select node with shortest distance to U (currently X) and determine shortest distance of its neighbors from U

- Node selected is min distance

Node	Distance Computation from U						Path
U	0	-	-				-
V	∞	2	2				U → V
W	∞	5	4				U → X → W
X	∞	1	-				U → X
Y	∞	∞	2				U → X → Y
Z	∞	∞	∞				-

Dijkstra's Algorithm Example (Node U)

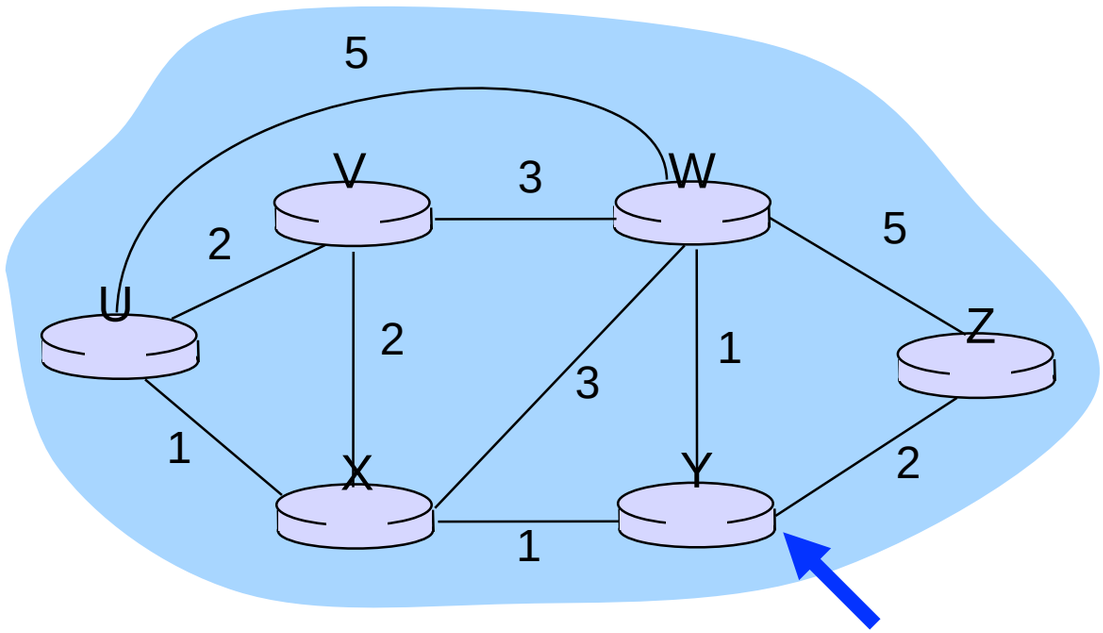


Select node with shortest distance to U (currently X) and determine shortest distance of its neighbors from U

- Node selected is min distance

Node	Distance Computation from U						Path
U	**0**	-	-	-			-
V	∞	2	**2**	-			**U ➔ V**
W	∞	5	4	4			U ➔ X ➔ W
X	∞	**1**	-	-			**U ➔ X**
Y	∞	∞	2	2			U ➔ X ➔ Y
Z	∞	∞	∞	∞			-

Dijkstra's Algorithm Example (Node U)

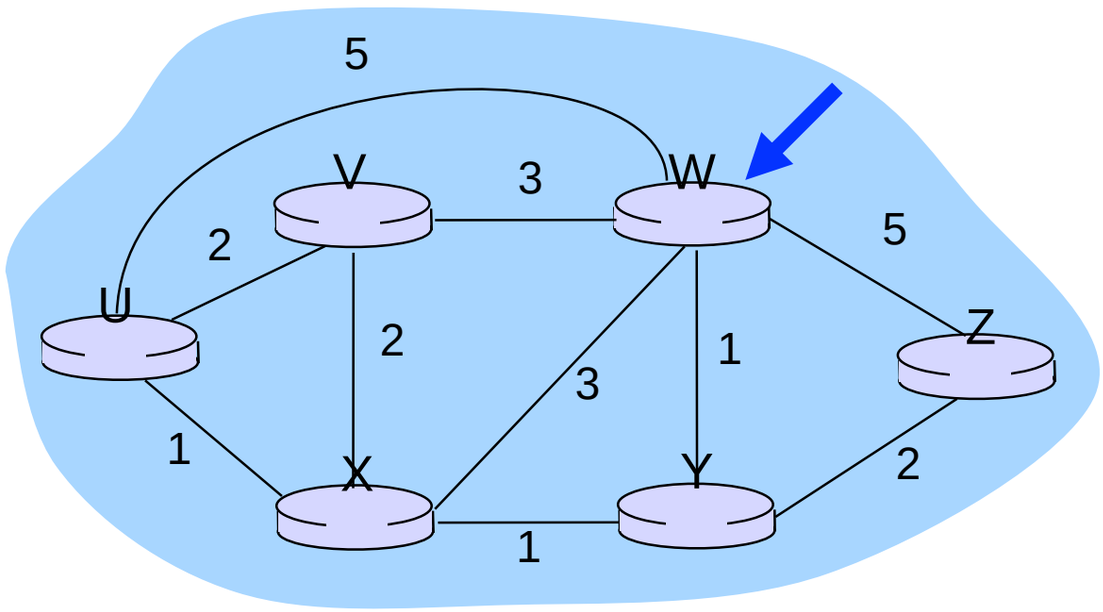


Select node with shortest distance to U (currently Y) and determine shortest distance of its neighbors from U

- Node selected is min distance

Node	Distance Computation from U						Path
U	0	-	-	-	-		-
V	∞	2	2	-	-		U → V
W	∞	5	4	4	3		U → X → Y → W
X	∞	1	-	-	-		U → X
Y	∞	∞	2	2	-		U → X → Y
Z	∞	∞	∞	∞	4		U → X → Y → Z

Dijkstra's Algorithm Example (Node U)

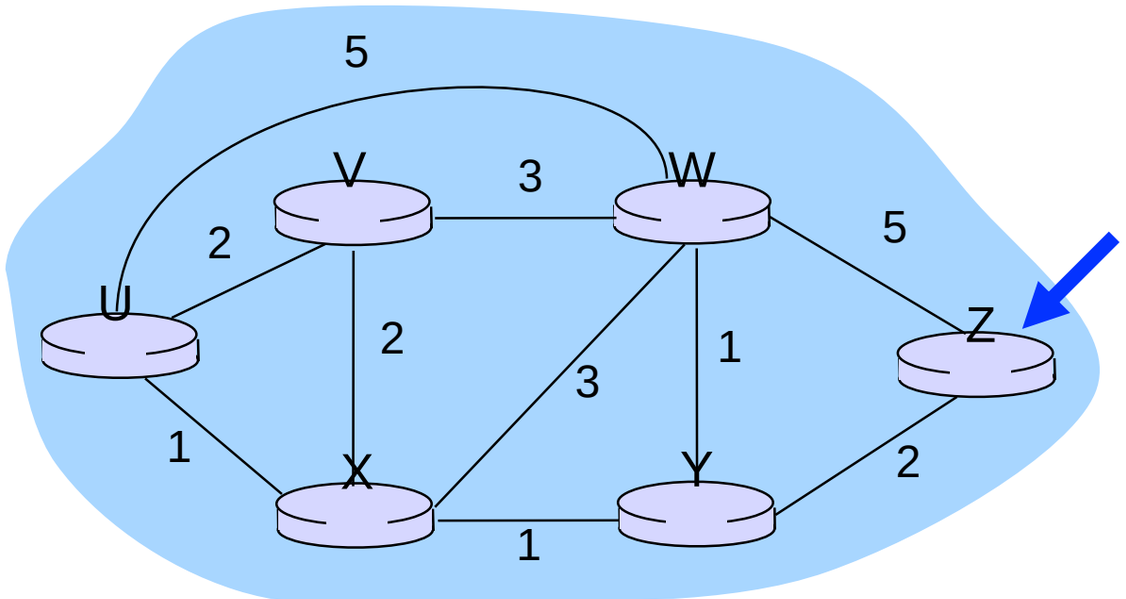


Select node with shortest distance to U (currently W) and determine shortest distance of its neighbors from U

- Node selected is min distance

Node	Distance Computation from U						Path
U	**0**	-	-	-	-	-	-
V	∞	2	**2**	-	-	-	**U ➔ V**
W	∞	5	4	4	**3**	-	**U ➔ X ➔ Y ➔ W**
X	∞	**1**	-	-	-	-	**U ➔ X**
Y	∞	∞	2	**2**	-	-	**U ➔ X ➔ Y**
Z	∞	∞	∞	∞	4	4	U ➔ X ➔ Y ➔ Z

Dijkstra's Algorithm Example (Node U)



Select node with shortest distance to U (currently Z) and determine shortest distance of its neighbors from U

- All nodes have been accounted for, so terminate

Node	Distance Computation from U						Path
U	**0**	-	-	-	-	-	-
V	∞	2	**2**	-	-	-	**U → V**
W	∞	5	4	4	**3**	-	**U → X → Y → W**
X	∞	**1**	-	-	-	-	**U → X**
Y	∞	∞	2	**2**	-	-	**U → X → Y**
Z	∞	∞	∞	∞	4	**4**	**U → X → Y → Z**