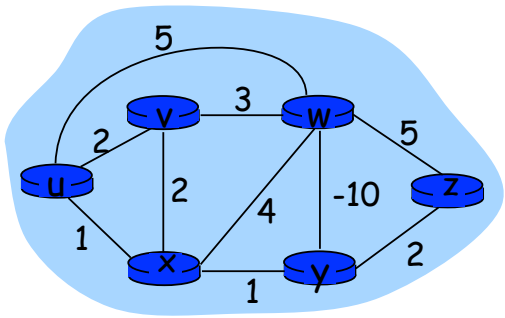
Network Routing Supplemental Problems

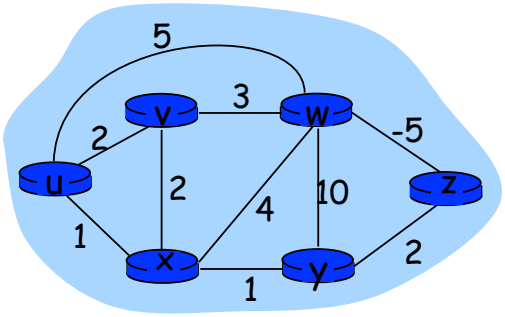
6 Nov 2012

Q1: Does Dijkstra's algorithm work on the network below?



A1: The question of shortest path makes no sense in a graph with negative weight loops/ cycles. This is because you can keep going round the negative cycle an infinite number of times, lowering the path cost arbitrarily.

Q2. Does Dijkstra's algorithm work on the network below?

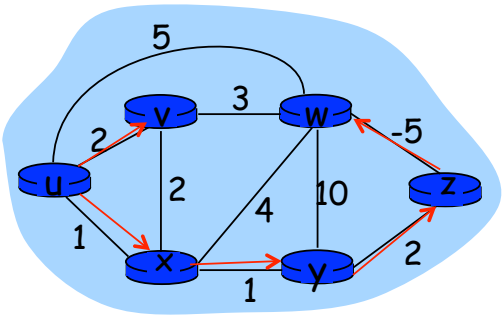


A1: Since there is not negative loop, there is a shortest path. However, since Dijkstra is greedy, it may not find it. Dijkstra does not account for the fact that a negative weight may come later.

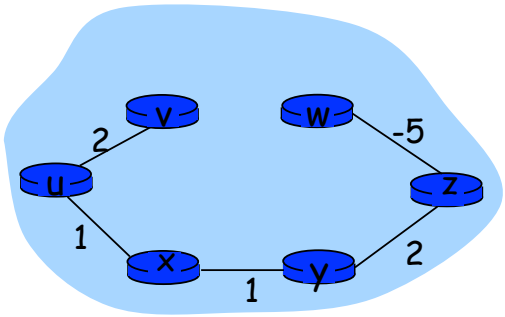
Solution to Q2

The shortest path from U to Z is U-X-W-Z with cost 0.

Dijsktra finds the following shortest path tree:

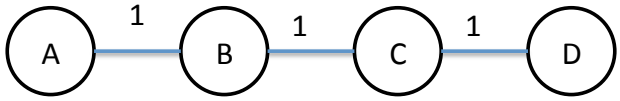


Dijkstra on the graph of Q2



The Dijkstra spanning tree

Q3: What is the count to infinity problem in distance vector routing?



Consider the network above. Assume that the network has stabilized.
Q3a: What are the routing tables at all nodes in the following format?
Q3b: What happens if the link between A and B is cut?

Routing table at A		
Dest.	Next Hop	Cost to dest.
B		
C		
D		

Routing Tables for Q3

To Dest	At A	At B	At C	At D
A	A/0	A/1	B/2	C/3
B	B/1	B/0	C/2	D/3
C	B/2	C/1	C/0	C/1
D	B/3	C/2	D/1	D/0

Count-to-infinity problem in Q3

Now imagine that the link between A and B is cut. At this time, B corrects its table. After a specific amount of time, routers exchange their tables, and so B receives C's routing table. Since C doesn't know what has happened to the link between A and B, it says that it has a link to A with the weight of 2 (1 for C to B, and 1 for B to A -- it doesn't know B has no link to A). B receives this table and thinks there is a separate link between C and A, so it corrects its table and changes infinity to 3 (1 for B to C, and 2 for C to A, as C said). Once again, routers exchange their tables. When C receives B's routing table, it sees that B has changed the weight of its link to A from 1 to 3, so C updates its table and changes the weight of the link to A to 4 (1 for C to B, and 3 for B to A, as B said). This process loops until all nodes find out that the weight of link to A is infinity.

Count-to-infinity problem in Q3

	B	C	D
Sum of weight to A after link cut	∞,A	2,B	3,C
Sum of weight to B after 1st updating	3,C	2,B	3,C
Sum of weight to A after 2nd updating	3,C	4,B	3,C
Sum of weight to A after 3rd updating	5,C	4,B	5,C
Sum of weight to A after 4th updating	5,C	6,B	5,C
Sum of weight to A after 5th updating	7,C	6,B	7,C
Sum of weight to A after nth updating	...	...	...
∞	∞	∞	∞