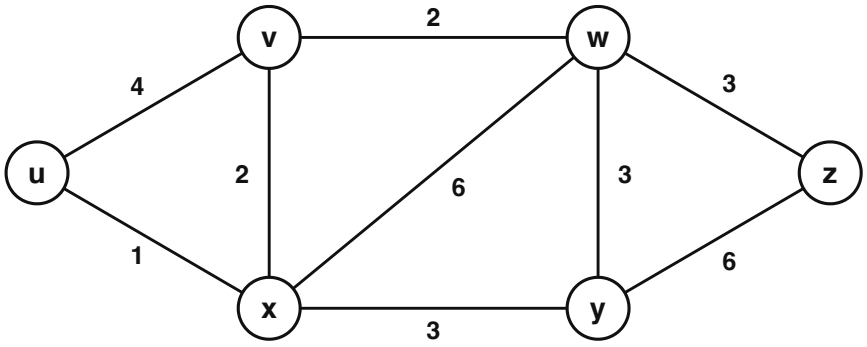
Q1 **Link-State Routing: Dijkstra** Ch. 5 slides 13–27

Names: _____

Router u runs Dijkstra's algorithm to find its least-cost path to every other router. Each number is the cost of one link, in both directions.



1. Run Dijkstra's algorithm from u. Leave a column blank once its router is in N′. Step 0 is done for you.

Step	N′	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	*u*	*4, u*	∞	*1, u*	∞	∞
1						
2						
3						
4						
5						

2. Darken the links of u's least-cost-path tree on the figure. Then fill in u's forwarding table.

Destination	v	w	x	y	z
Outgoing link					

3. Right after u builds its table, link v–x fails. u learns of the failure from a flooded link-state advertisement and reruns Dijkstra. Which destinations in u's forwarding table now use a different outgoing link, and what do they cost now?

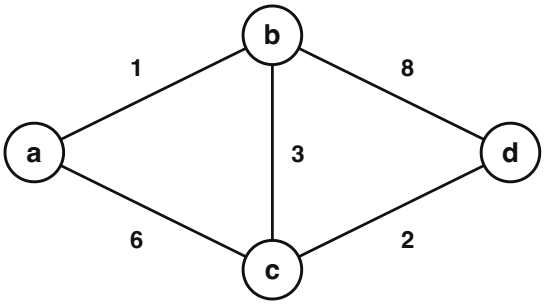
Discussion

In your answer, u sends traffic for every destination out the same link. Suppose instead each link's cost were its current traffic load. What happens the next time u reruns Dijkstra? And the time after that?

v261028-1247

Q2 **Distance Vector: Bellman-Ford** Ch. 5 slides 31–34, 43–48

Four routers run the distance vector algorithm. At **t = 0** each router knows only the costs of its own links, and sends that distance vector to its neighbors. The table holds the vectors sent at t = 0.



Vector at t = 0	to a	to b	to c	to d
D_a	0	1	6	∞
D_b	1	0	3	8
D_c	6	3	0	2
D_d	∞	8	2	0

1. Write the Bellman-Ford equation a uses to compute $D_a(d)$.

$D_a(d)$ = _____

2. At **t = 1**, a receives the t = 0 vectors from b and c. Compute a's new distance vector and next hops.

Destination	via b: c(a,b) + D_b	via c: c(a,c) + D_c	D_a	Next hop
b				
c				
d				

By **t = 2** the vectors have stopped changing. These are the vectors every router holds at t = 2:

Vector at t = 2	to a	to b	to c	to d
D_a	0	1	4	6
D_b	1	0	3	5
D_c	4	3	0	2
D_d	6	5	2	0

Discussion

Right after t = 2, link c–d fails. c knows only its own link costs and the vectors its neighbors sent it, which are the rows in the t = 2 table. Using only that information, what does c compute for its cost to d, and what is wrong with that answer? Would a link-state router make the same mistake?