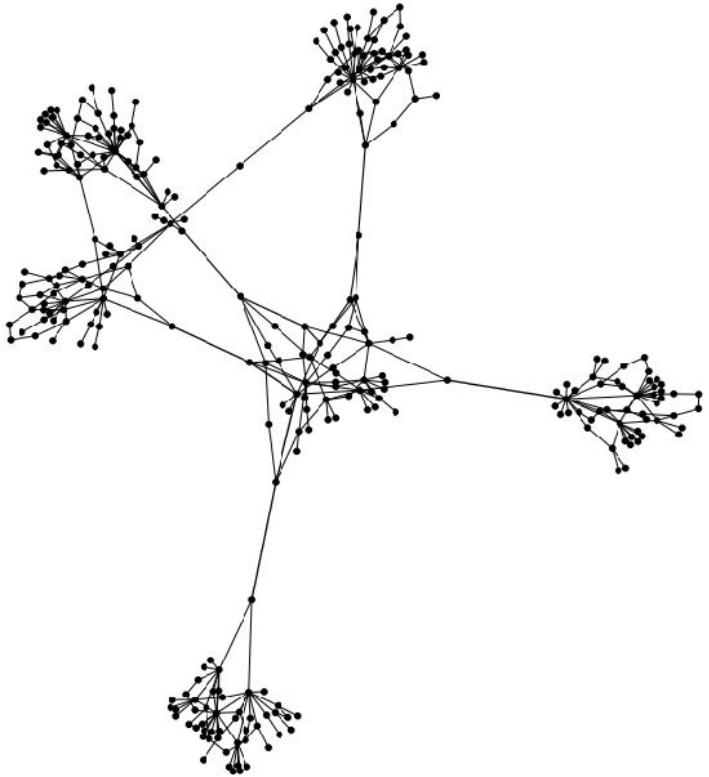
Recitation 10

Internet Technology (Section 01)

Routing Protocols

- Establishing routes through a graph of network devices.
- Metrics
  - Latency
  - # of hops
  - Reliability
  - Bandwidth



What the routing algorithm builds

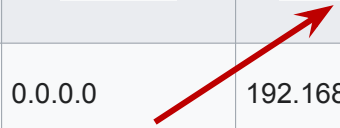
Example Routing Table

Network destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.100	10
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.100	192.168.0.100	10
192.168.0.100	255.255.255.255	127.0.0.1	127.0.0.1	10
192.168.0.1	255.255.255.255	192.168.0.100	192.168.0.100	10

What the routing algorithm builds

Example Routing Table

Network destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.100	10
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.100	192.168.0.100	10
192.168.0.100	255.255.255.255	127.0.0.1	127.0.0.1	10
192.168.0.1	255.255.255.255	192.168.0.100	192.168.0.100	10



What the routing algorithm builds

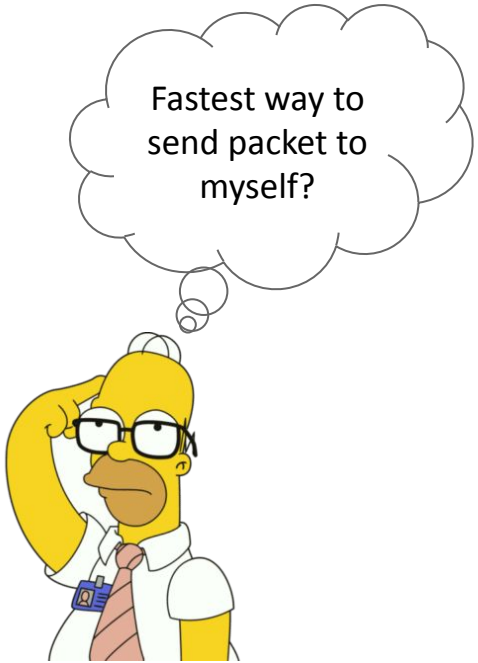
Example Routing Table

Minimize

Next Hop

Network destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.100	10
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.100	192.168.0.100	10
192.168.0.100	255.255.255.255	127.0.0.1	127.0.0.1	10
192.168.0.1	255.255.255.255	192.168.0.100	192.168.0.100	10

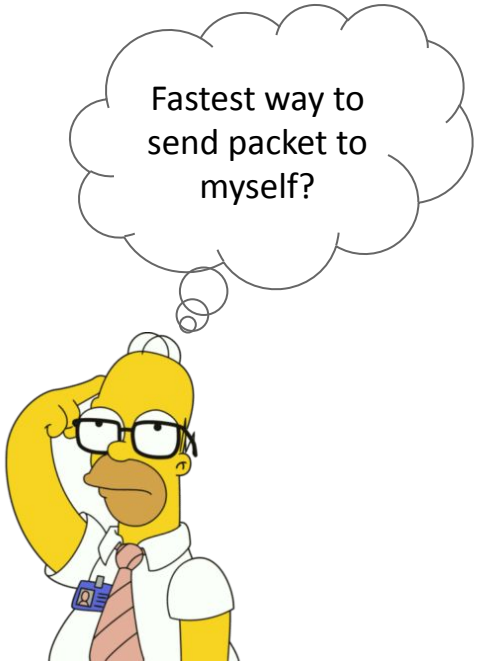
What the routing algorithm builds



Example Routing Table

Network destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.100	10
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.100	192.168.0.100	10
192.168.0.100	255.255.255.255	127.0.0.1	127.0.0.1	10
192.168.0.1	255.255.255.255	192.168.0.100	192.168.0.100	10

What the routing algorithm builds



Example Routing Table

Network destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.100	10
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.100	192.168.0.100	10
192.168.0.100	255.255.255.255	127.0.0.1	127.0.0.1	10
192.168.0.1	255.255.255.255	192.168.0.100	192.168.0.100	10

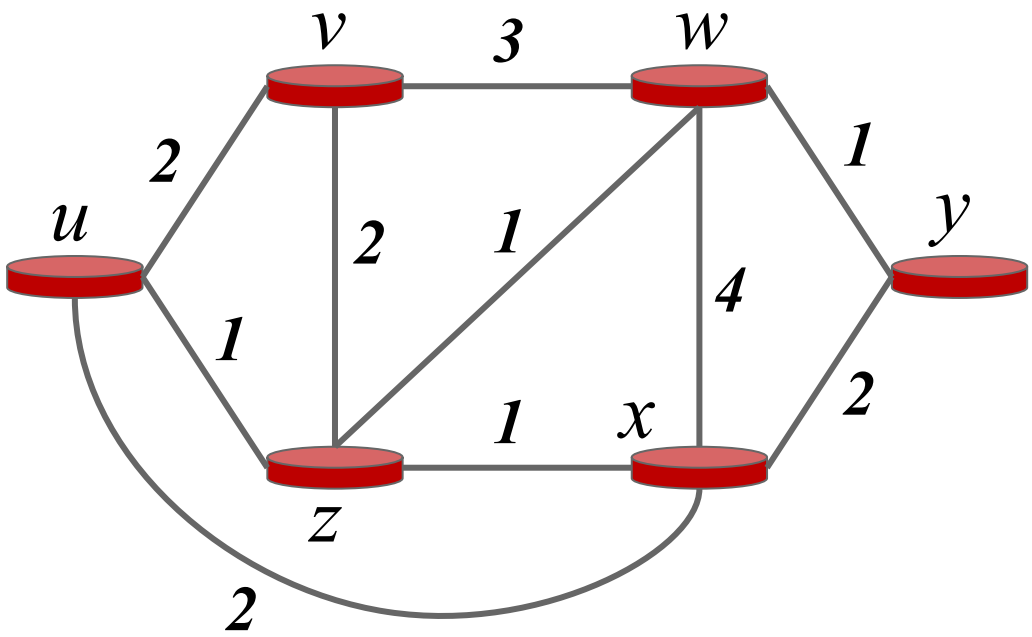
What the routing algorithm builds

Example Routing Table

Network destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.100	10
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	**1**
192.168.0.0	255.255.255.0	192.168.0.100	192.168.0.100	10
192.168.0.100	255.255.255.255	127.0.0.1	127.0.0.1	**10**
192.168.0.1	255.255.255.255	192.168.0.100	192.168.0.100	10

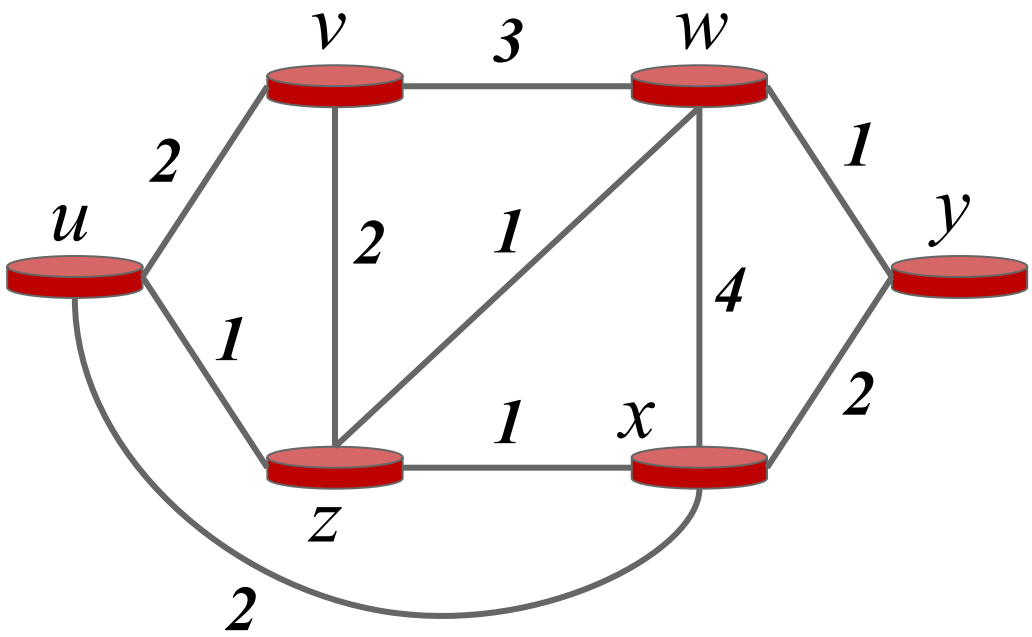
Building routing tables

- Consider graph *G* with edge weights representing hop count.
- Routing table is built by running shortest-path algorithms on *G*.



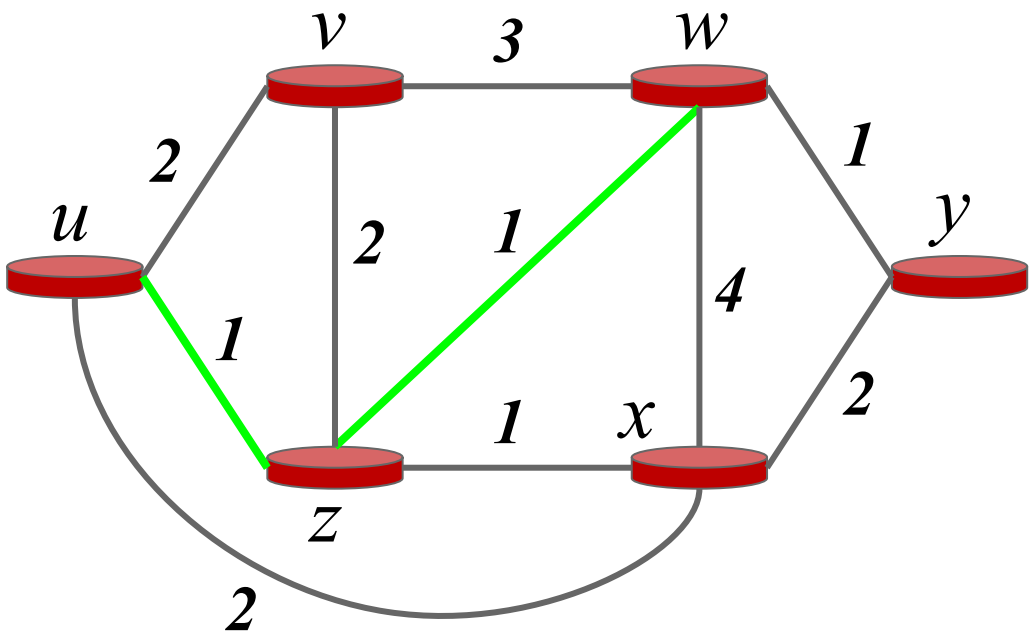
Building routing tables

- How to get from *u* to *w* with least number of hops?



Building routing tables

- How to get from *u* to *w* with least number of hops?

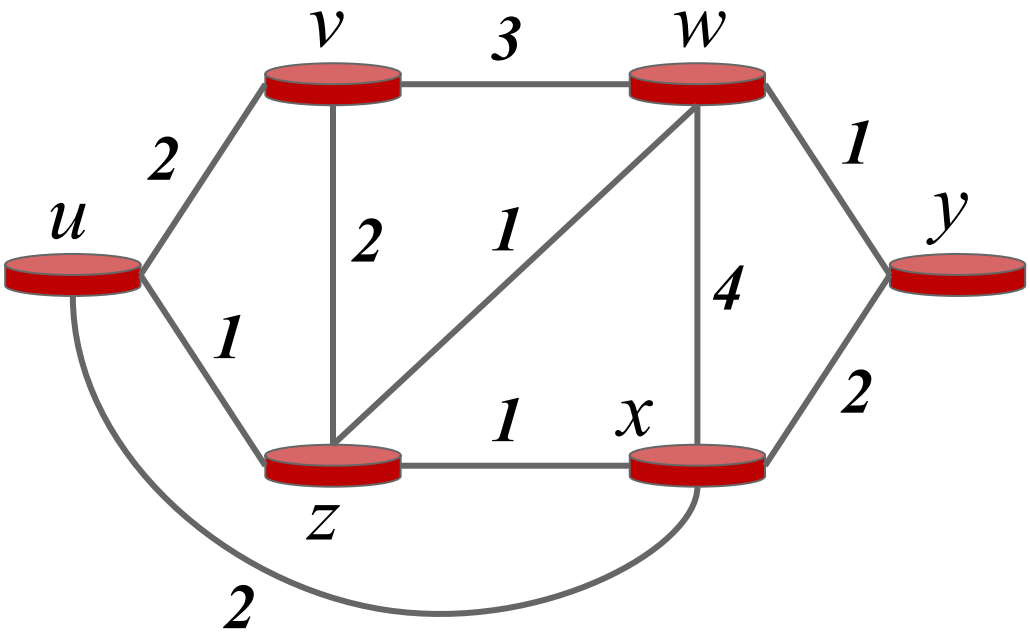


Link-State Routing

- A class of routing protocols to build routing tables.
- Stage 1
  - Each routing node builds network graph.
- Stages 2
  - Produce routing tables from graph.

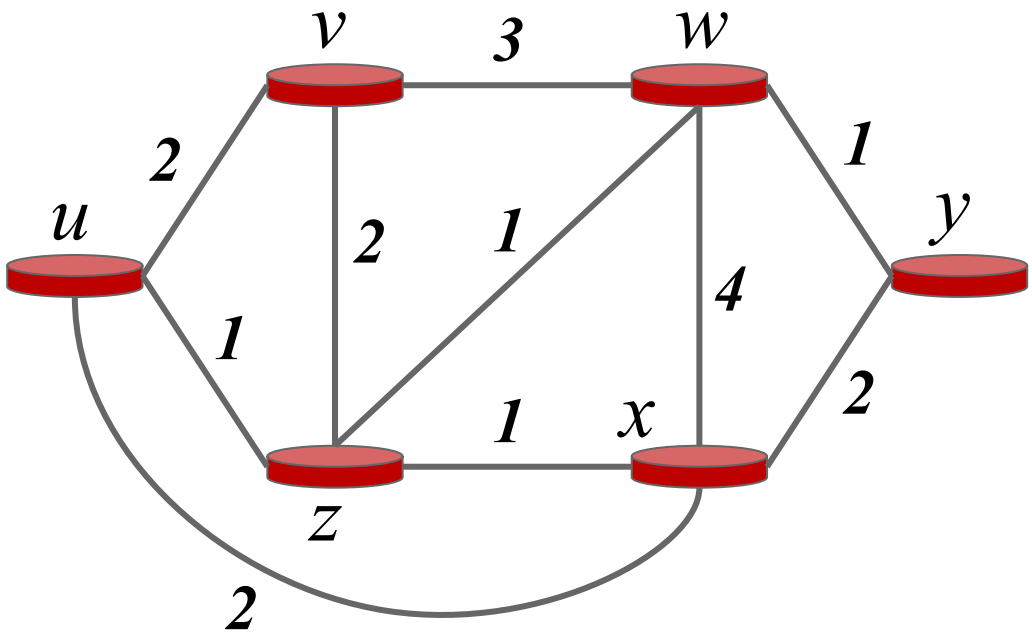
Dijkstra's Shortest Path Algorithm

Round	u	v	z	w	x	y
1						
2						
3						
4						
5						
6						



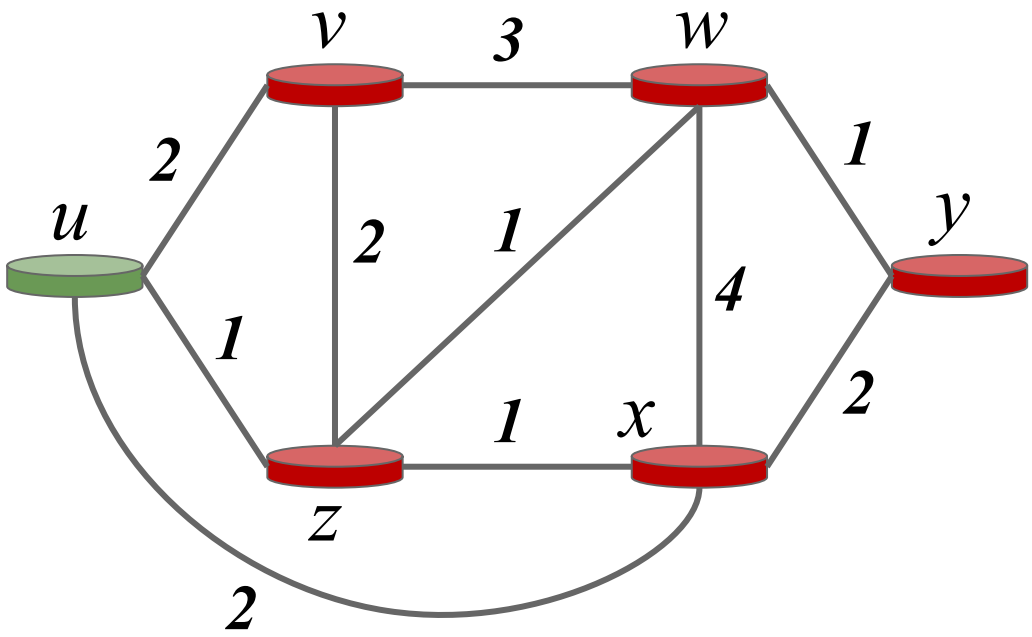
Dijkstra's Shortest Path Algorithm

Round	u	v	z	w	x	y
1	0, -	∞, -	∞, -	∞, -	∞, -	∞, -
2						
3						
4						
5						
6						



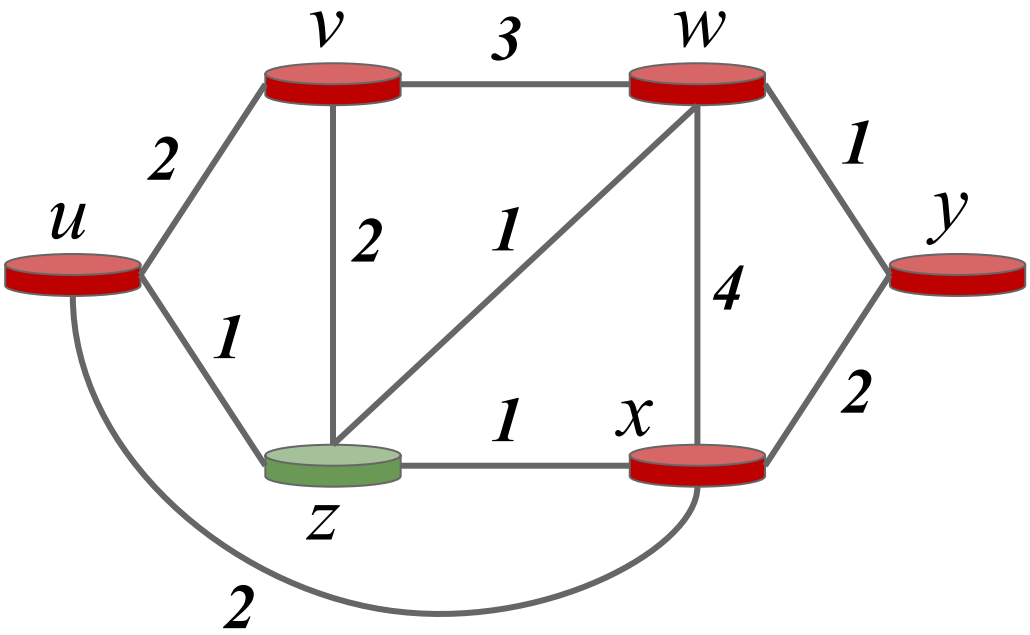
Dijkstra's Shortest Path Algorithm

Round	u	v	z	w	x	y
1	0, -	∞, -	∞, -	∞, -	∞, -	∞, -
2		2, u	1, u	∞, -	2, u	∞, -
3						
4						
5						
6						



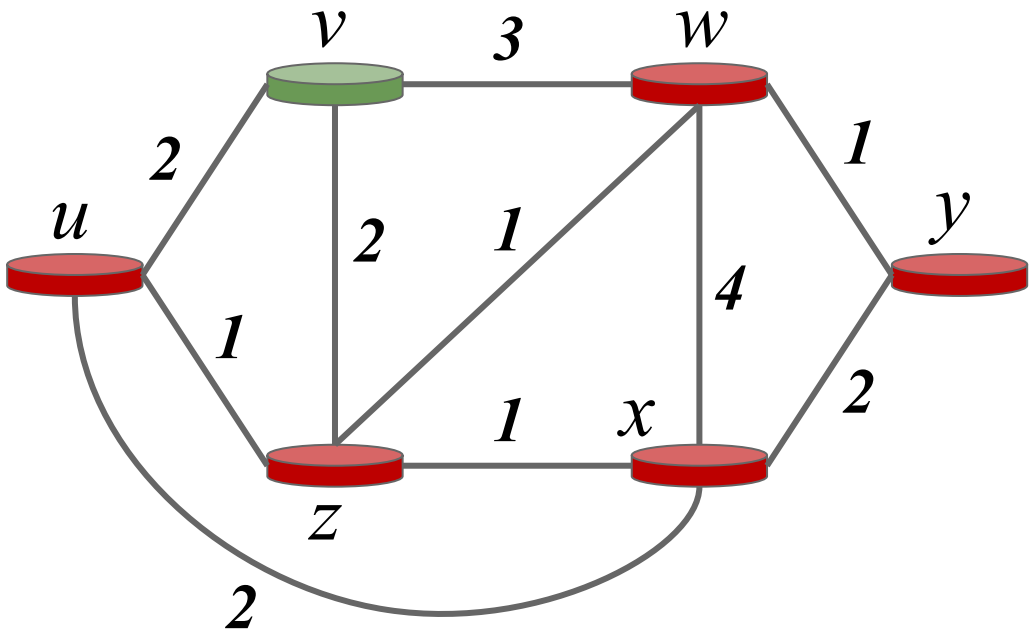
Dijkstra's Shortest Path Algorithm

Round	*u*	*v*	*z*	*w*	*x*	*y*
1	0, -	∞, -	∞, -	∞, -	∞, -	∞, -
2		2, u	1, u	∞, -	2, u	∞, -
3		2, u		2, z	2, u	∞, -
4						
5						
6						



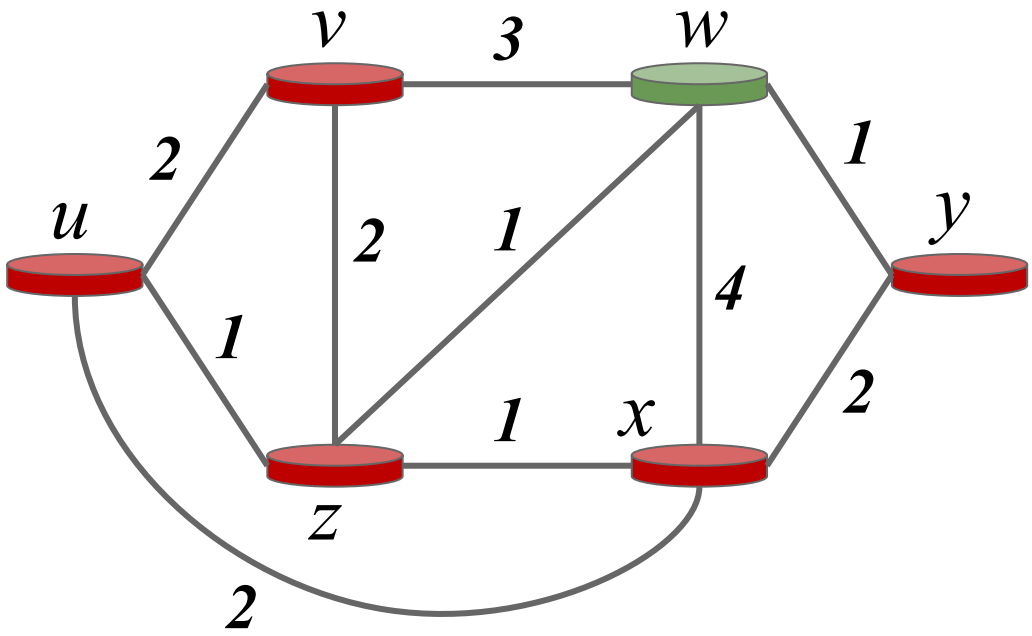
Dijkstra's Shortest Path Algorithm

Round	*u*	*v*	*z*	*w*	*x*	*y*
1	0, -	∞, -	∞, -	∞, -	∞, -	∞, -
2		2, u	1, u	∞, -	2, u	∞, -
3		2, u		2, z	2, u	∞, -
4				2, z	2, u	∞, -
5						
6						



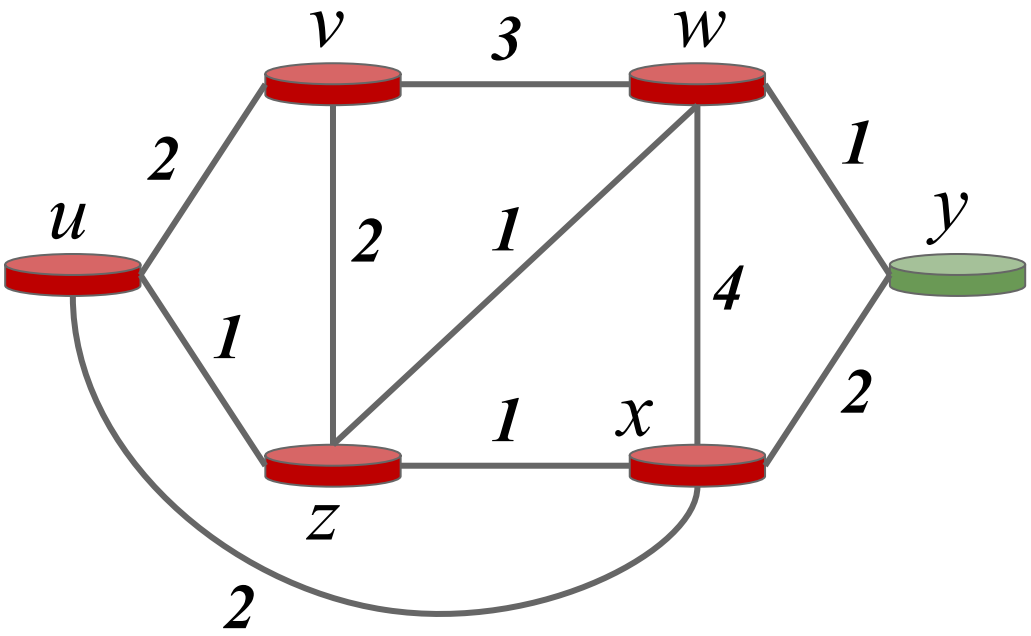
Dijkstra's Shortest Path Algorithm

Round	*u*	*v*	*z*	*w*	*x*	*y*
1	0, -	∞, -	∞, -	∞, -	∞, -	∞, -
2		2, u	1, u	∞, -	2, u	∞, -
3		2, u		2, z	2, u	∞, -
4				2, z	2, u	∞, -
5					2, u	3, w
6						



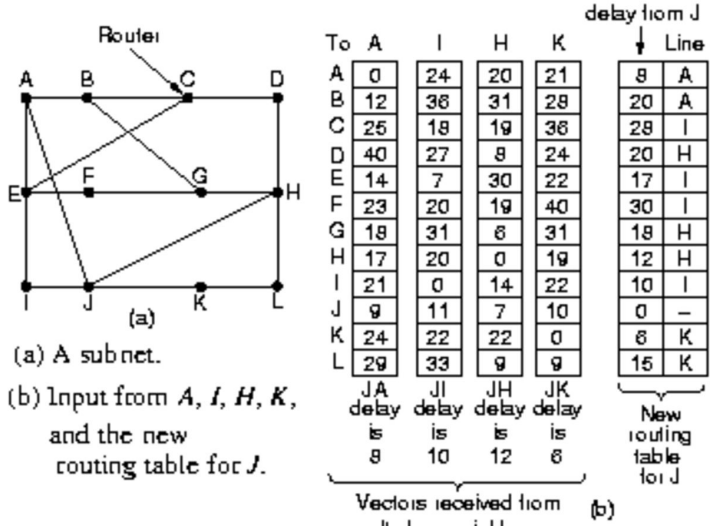
Dijkstra's Shortest Path Algorithm

Round	*u*	*v*	*z*	*w*	*x*	*y*
1	0, -	∞, -	∞, -	∞, -	∞, -	∞, -
2		2, u	1, u	∞, -	2, u	∞, -
3		2, u		2, z	2, u	∞, -
4				2, z	2, u	∞, -
5					2, u	3, w
6						3, w

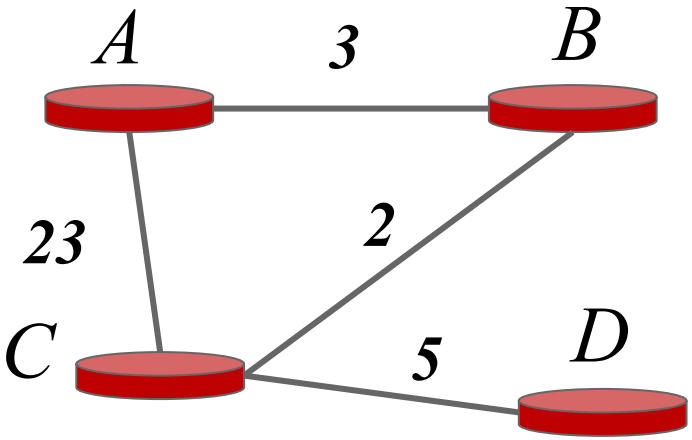


Distance-vector Routing

- Another class of routing protocols to build routing tables.
- Idea: Each nodes advertises only its distance value (DV) to other nodes.
  - Advertises occur until all routers converge to similar DVs



Distance-vector Routing Example



T=0

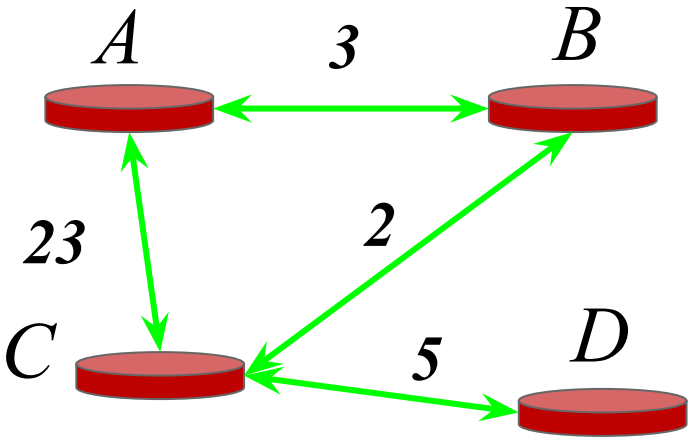
from A	via A	via B	via C	via D
to A				
to B		3		
to C			23	
to D				

from B	via A	via B	via C	via D
to A	3			
to B				
to C			2	
to D				

from C	via A	via B	via C	via D
to A	23			
to B		2		
to C				
to D				5

from D	via A	via B	via C	via D
to A				
to B				
to C			5	
to D				

Distance-vector Routing Example



T=0

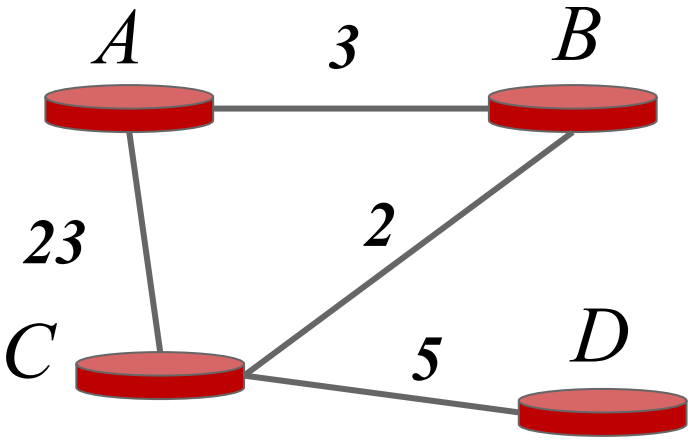
from A	via A	via B	via C	via D
to A				
to B		3		
to C			23	
to D				

from B	via A	via B	via C	via D
to A	3			
to B				
to C			2	
to D				

from C	via A	via B	via C	via D
to A	23			
to B		2		
to C				
to D				5

from D	via A	via B	via C	via D
to A				
to B				
to C			5	
to D				

Distance-vector Routing Example



T=1

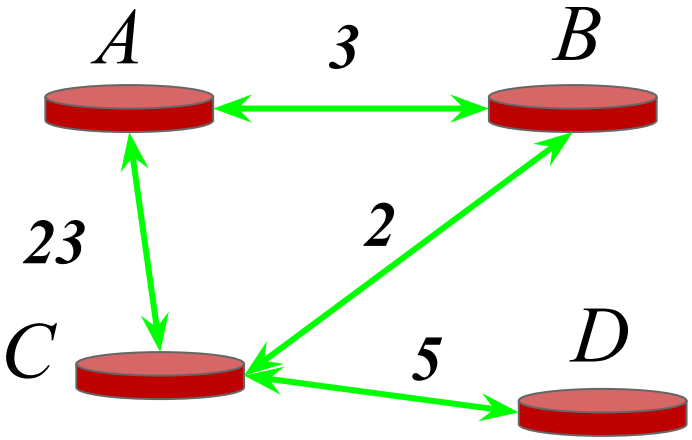
from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D			28	

from B	via A	via B	via C	via D
to A	3		25	
to B				
to C	26		2	
to D			7	

from C	via A	via B	via C	via D
to A	23	5		
to B	26	2		
to C				
to D				5

from D	via A	via B	via C	via D
to A			28	
to B			7	
to C			5	
to D				

Distance-vector Routing Example



T=1

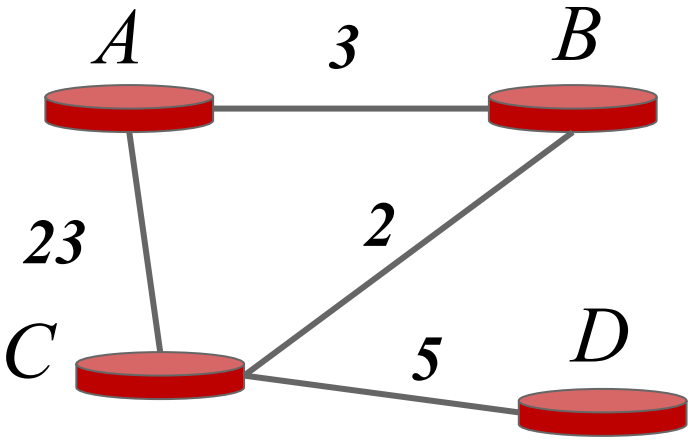
from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D			28	

from B	via A	via B	via C	via D
to A	3		25	
to B				
to C	26		2	
to D			7	

from C	via A	via B	via C	via D
to A	23	5		
to B	26	2		
to C				
to D				5

from D	via A	via B	via C	via D
to A			28	
to B			7	
to C			5	
to D				

Distance-vector Routing Example



T=2

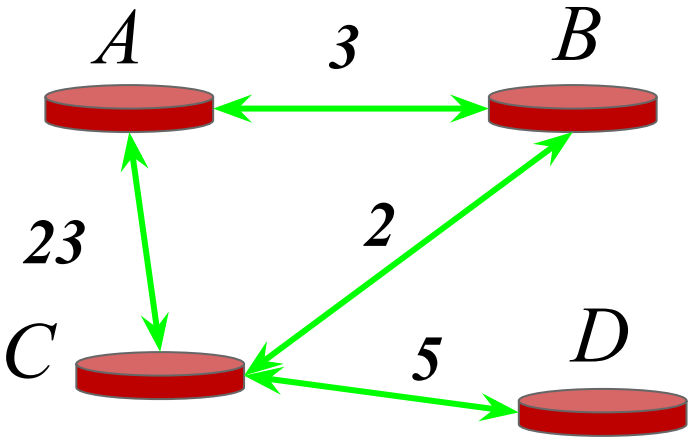
from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D		10	28	

from B	via A	via B	via C	via D
to A	3		7	
to B				
to C	8		2	
to D	31		7	

from C	via A	via B	via C	via D
to A	23	5		33
to B	26	2		12
to C				
to D	51	9		5

from D	via A	via B	via C	via D
to A			10	
to B			7	
to C			5	
to D				

Distance-vector Routing Example



T=2

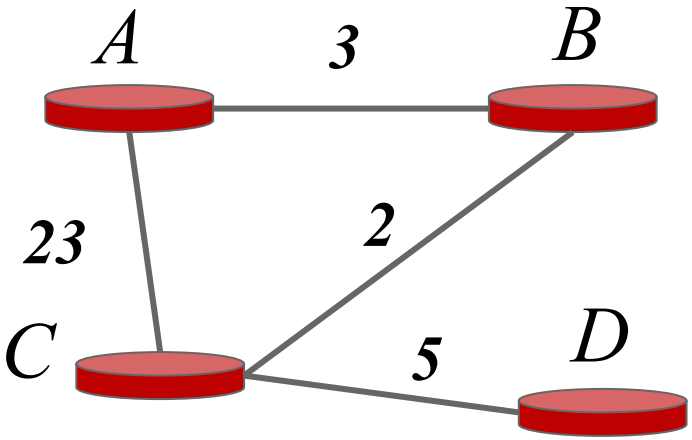
from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D		10	28	

from B	via A	via B	via C	via D
to A	3		7	
to B				
to C	8		2	
to D	31		7	

from C	via A	via B	via C	via D
to A	23	5		33
to B	26	2		12
to C				
to D	51	9		5

from D	via A	via B	via C	via D
to A			10	
to B			7	
to C			5	
to D				

Distance-vector Routing Example



T=3

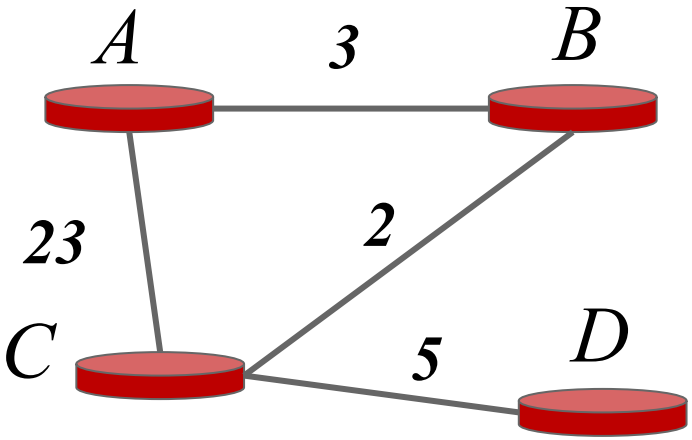
from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D		10	28	

from B	via A	via B	via C	via D
to A	3		7	
to B				
to C	8		2	
to D	13		7	

from C	via A	via B	via C	via D
to A	23	5		15
to B	26	2		12
to C				
to D	33	9		5

from D	via A	via B	via C	via D
to A			10	
to B			7	
to C			5	
to D				

Distance-vector Routing Example



We just did the Bellman–Ford algorithm!

T=3

from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D		10	28	

from B	via A	via B	via C	via D
to A	3		7	
to B				
to C	8		2	
to D	13		7	

from C	via A	via B	via C	via D
to A	23	5		15
to B	26	2		12
to C				
to D	33	9		5

from D	via A	via B	via C	via D
to A			10	
to B			7	
to C			5	
to D				