Exercise 5

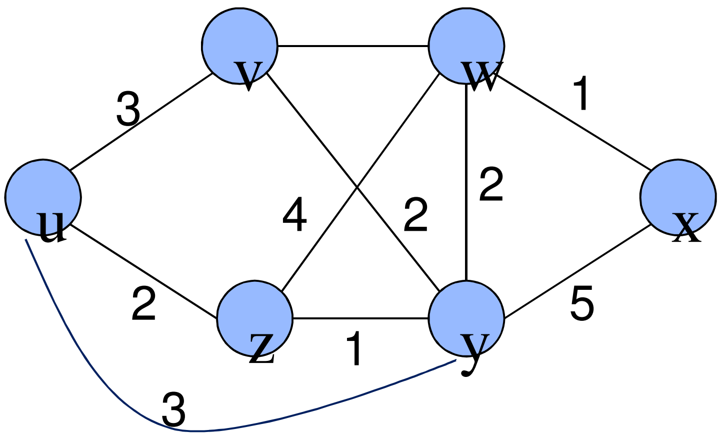
Narisu Tao

narisu.tao@informatik.uni-goettingen.de

computer NETWORKS

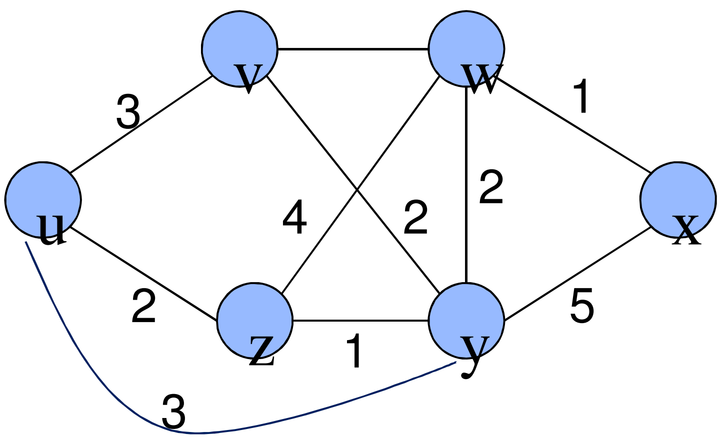
Dijkstra's algorithm

- Q1: Given the following network, use Dijkstra's algorithm to find the least cost paths from node u. Please provide a table showing the steps of the algorithm, a graph showing the resulting shortest-path tree from u and the final forwarding table of u.



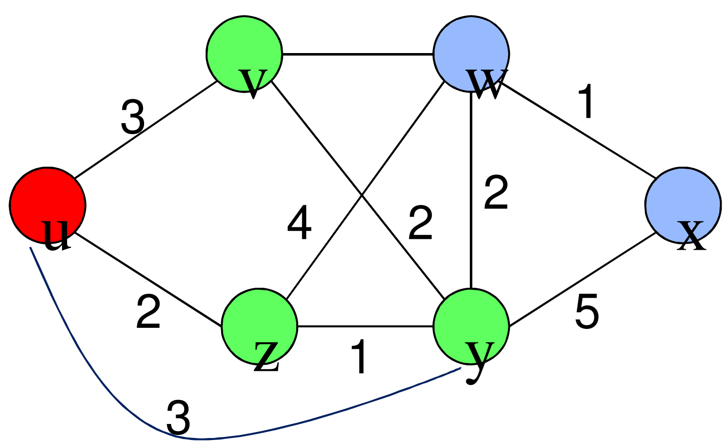
Dijkstra's algorithm (cont’d)

Step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)



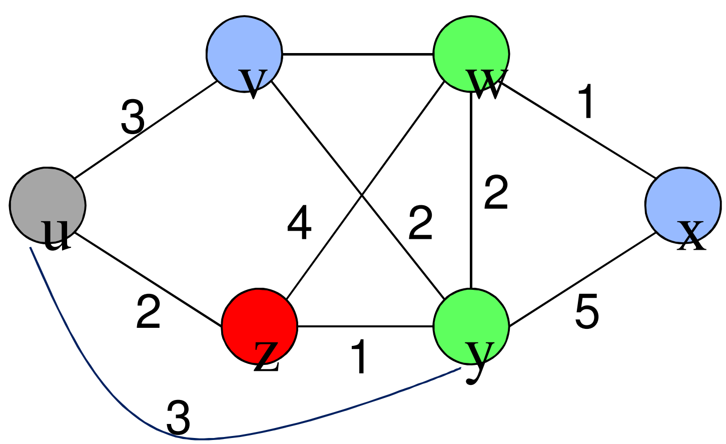
Dijkstra's algorithm (cont'd)

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



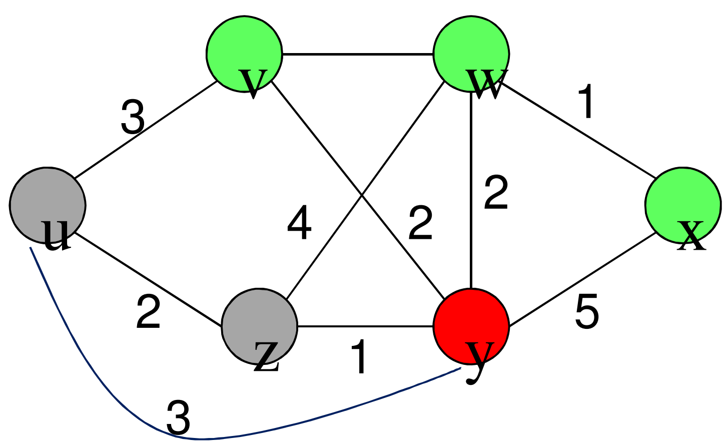
Dijkstra's algorithm (cont'd)

Step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	3,u	∞	∞	3,u	2,u
1	uz	3,u	6,z	∞	3,u	
2						
3						
4						
5						



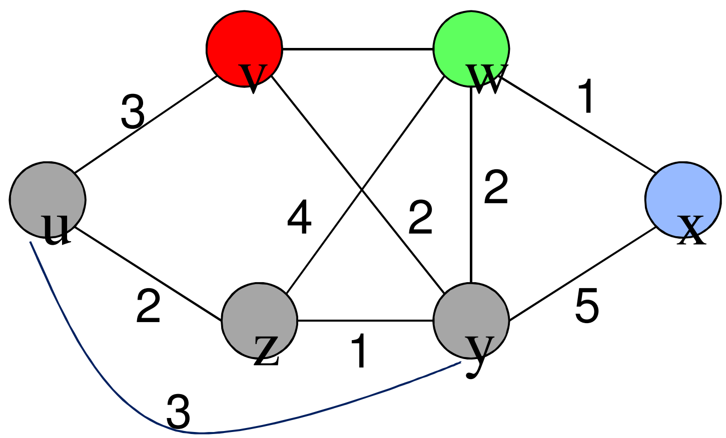
Dijkstra's algorithm (cont’d)

Step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	3,u	∞	∞	3,u	2,u
1	uz	3,u	6,z	∞	3,u	
2	uzy	3,u	5,y	8,y		
3						
4						
5						



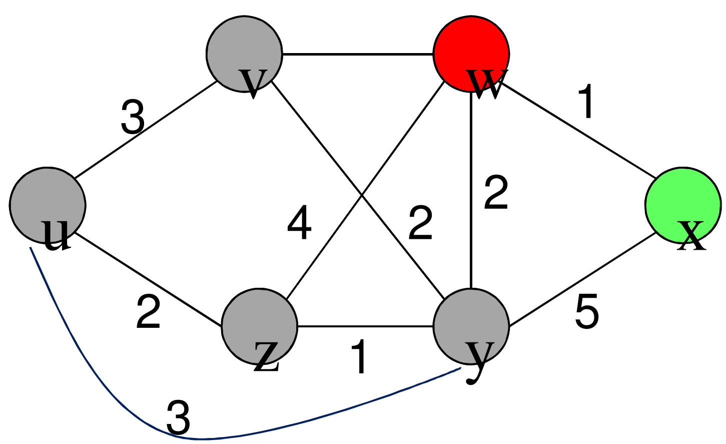
Dijkstra's algorithm (cont'd)

Step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	3,u	∞	∞	3,u	2,u
1	uz	3,u	6,z	∞	3,u	
2	uzy	3,u	5,y	8,y		
3	uzyv		5,y	6,w		
4						
5						



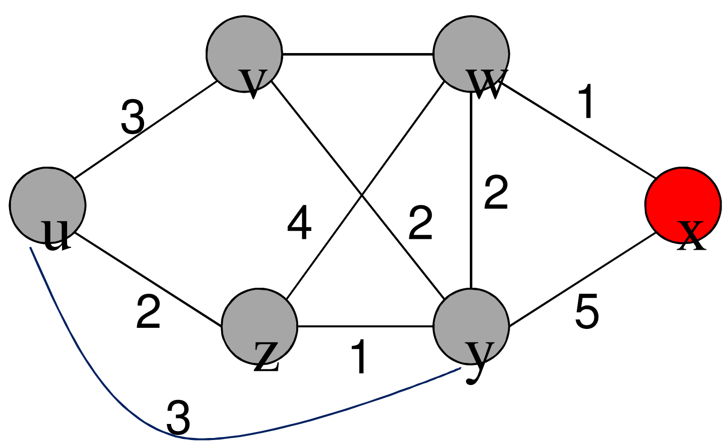
Dijkstra's algorithm (cont’d)

Step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	3,u	∞	∞	3,u	2,u
1	uz	3,u	6,z	∞	3,u	
2	uzy	3,u	5,y	8,y		
3	uzyv		5,y	6,w		
4	uzyvw			6,w		
5						

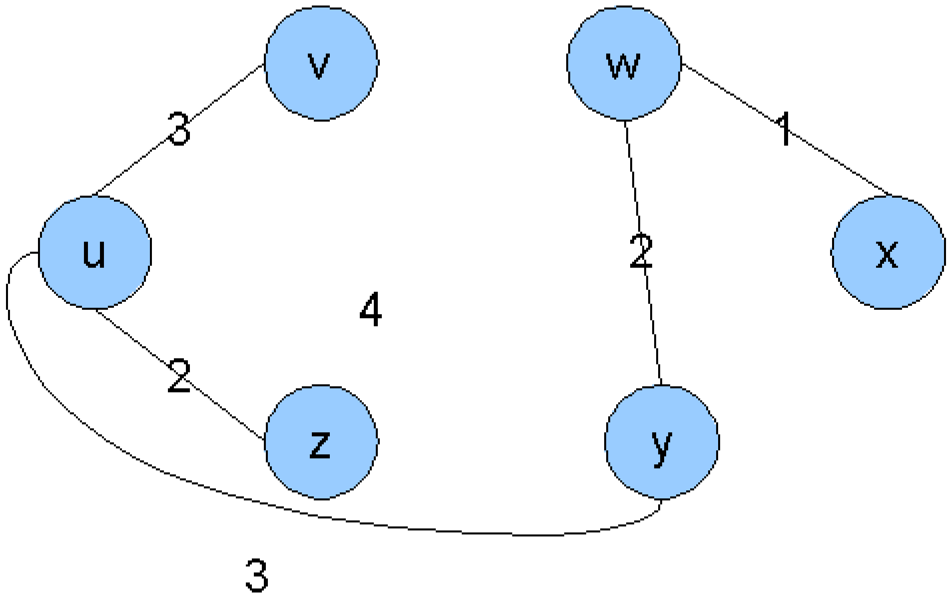


Dijkstra's algorithm (cont'd)

Step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	3,u	∞	∞	3,u	2,u
1	uz	3,u	6,z	∞	3,u	
2	uzy	3,u	5,y	8,y		
3	uzyv		5,y	6,w		
4	uzyvw			6,w		
5	uzyvwx					



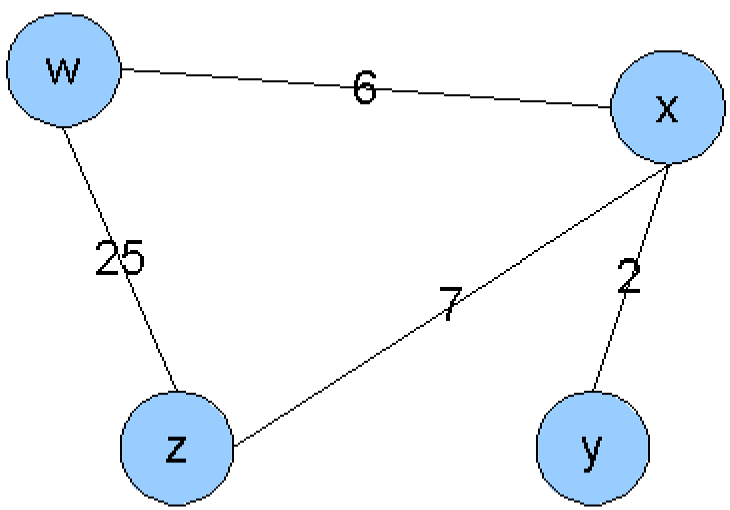
Dijkstra's algorithm (cont’d)



Dest.	Link.
z	z
y	y
v	v
w	y
x	y

Distance Vector algorithm

- Q2: Given the following network, use the Distance Vector algorithm to find the least cost paths for all nodes. Fill the provided tables and indicate with arrows between the tables when a node sends a distance vector to another node.



Distance Vector algorithm

Node w		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node w		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node w		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node w		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node x		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node x		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node x		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node x		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node y		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node y		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node y		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node y		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node z		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node z		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node z		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Node z		cost to			
		w	x	y	z
from	w				
	x				
	y				
	z				

Distance Vector algorithm

Node w	cost to			
	w	x	y	z
from w	**0**	**6**	∞	**25**
from x	∞	∞	∞	∞
from y	∞	∞	∞	∞
from z	∞	∞	∞	∞

Node w	cost to			
	w	x	y	z
from w	**0**	**6**	**8**	**13**
from x	6	0	2	7
from y	∞	∞	∞	∞
from z	25	7	∞	0

Node w	cost to			
	w	x	y	z
from w	0	6	8	13
from x	6	0	2	7
from y	∞	∞	∞	∞
from z	13	7	9	0

Node w	cost to			
	w	x	y	z
from w				3
from x				
from y				
from z				

Node x	cost to			
	w	x	y	z
from w	∞	∞	∞	∞
from x	**6**	**0**	**2**	**7**
from y	∞	∞	∞	∞
from z	∞	∞	∞	∞

Node x	cost to			
	w	x	y	z
from w	0	6	∞	25
from x	6	0	2	7
from y	∞	2	0	∞
from z	25	7	∞	0

Node x	cost to			
	w	x	y	z
from w	0	6	8	13
from x	6	0	2	7
from y	8	2	0	9
from z	13	7	9	0

Node x	cost to			
	w	x	y	z
from w				
from x				
from y				
from z				

Node y	cost to			
	w	x	y	z
from w	∞	∞	∞	∞
from x	∞	∞	∞	∞
from y	∞	**2**	**0**	∞
from z	∞	∞	∞	∞

Node y	cost to			
	w	x	y	z
from w	∞	∞	∞	∞
from x	6	0	2	7
from y	**8**	**2**	**0**	**9**
from z	∞	∞	∞	∞

Node y	cost to			
	w	x	y	z
from w	∞	∞	∞	∞
from x	6	0	2	7
from y	8	2	0	9
from z	∞	∞	∞	∞

Node y	cost to			
	w	x	y	z
from w				
from x				
from y				
from z				

Node z	cost to			
	w	x	y	z
from w	∞	∞	∞	∞
from x	∞	∞	∞	∞
from y	∞	∞	∞	∞
from z	**25**	**7**	∞	**0**

Node z	cost to			
	w	x	y	z
from w	0	6	∞	25
from x	6	0	2	7
from y	∞	∞	∞	∞
from z	**13**	**7**	**9**	**0**

Node z	cost to			
	w	x	y	z
from w	0	6	8	13
from x	6	0	2	7
from y	∞	∞	∞	∞
from z	13	7	9	0

Node z	cost to			
	w	x	y	z
from w				
from x				
from y				
from z				

Comparison LV vs. DV

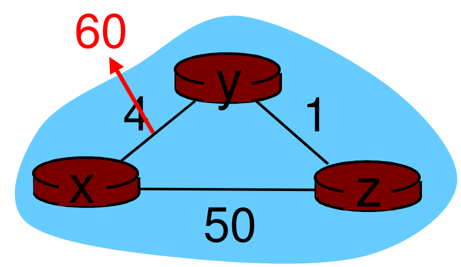
- **Q3:Compare Link State routing algorithms to Distance Vector algorithms in terms of scalability and robustness.**

- Scalability
  - LS uses broadcasts to disseminate complete knowledge about all links to entire network
  - DV only sends (local) information to neighboring nodes. Convergence time and DV size still increase with network size
- Robustness
  - LS: every router does its own calculations
  - DV: wrong DV will be used by neighboring nodes and further propagate the error

computer NETWORKS

Count-to-infinity problem

- **Q4: Explain the count-to-infinity problem using a simple example. How can this problem be avoided?**



- Link cost changes:

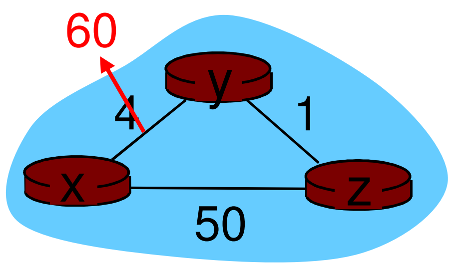
  44 iterations before algorithm stabilizes: see textbook

Dy(x) = min(c(y,x) + Dx(x), c(y,z)+Dz(x))
= min(4 + 0 , 1 + 5)
= min(4 , 6)
= 4

Dy(z) = min(c(y,x) + Dx(z), c(y,z)+Dz(z))
= min(4 + 5 , 1 + 0)
= min(9 , 1)
= 1

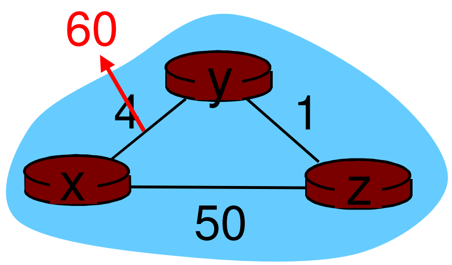
Dz(x) = min(c(z,x) + Dx(x), c(z,y)+Dy(x))
= min(50 + 0 , 1 + 4)
= min(50 , 5)
= 5

Dz(y) = min(c(z,x) + Dx(y), c(z,y)+Dy(y))
= min(50 + 4 , 1 + 0)
= min(54 , 1)
= 1



Dy(x) = min(c(y,x) + Dx(x), c(y,z)+Dz(x))
= min(**60** + 0 , 1 + 5)
= min(**60** , 6)
= **6**

Dz(x) = min(c(z,x) + Dx(x), c(z,y)+**Dy(x)**)
= min(50 + 0 , 1 + **6**)
= min(50 , **7**)
= 7



⋮

Dy(x) = min(c(y,x) + Dx(x), c(y,z)+Dz(x))
= min(**60** + 0 , 1 + **49**)
= min(**60** , **50**)
= **50**

Dz(x) = min(c(z,x) + Dx(x), c(z,y)+**Dy(x)**)
= min(50 + 0 , 1 + **50**)
= min(50 , **51**)
= 50

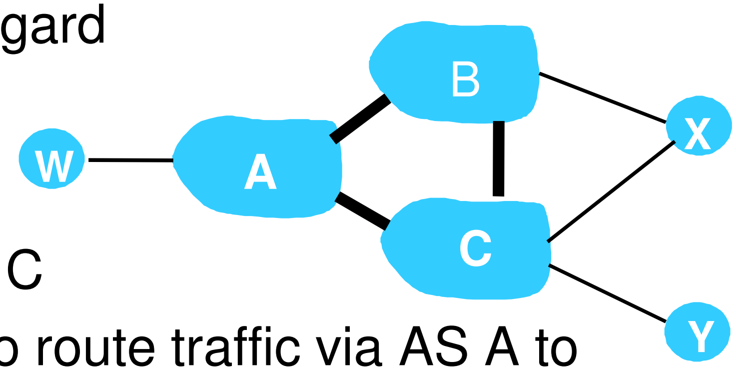
Count-to-infinity problem (con't)

- Count-to-infinitiy problem can be avoided using the poisoned reverse technique.
  - Router A will advertise a distance as infinite to Router B if Router B is on the advertised path
  - In the example: In its advertisements to Router B, Router C will advertise the cost to reach Router A as infinite as long as it routes packets to A via B
  - Poisoned reverse will only prevent routing loops that involve just two gateways. It is still possible to end up with patterns in which three gateways are engaged in mutual deception. E.g. A may believe it has a route through B, B through C, and C through A.

Routing policies

- Q5: How are routing policies used in BGP. Give one example.

- Routing policies determine …
  - … which BGP advertisements to regard
  - … which routes to advertise
- Example
  - AS A is connected to AS B and AS C
  - Policy : AS A does not want AS B to route traffic via AS A to AS C
  - Therefore, AS A does not advertise any route to reach AS C to AS B



Intra- vs. inter-AS routing

- Q6 :What is the difference between Intra-AS and Inter-AS routing? Why are different inter-AS and intra-AS protocols used in the Internet? ? Name one example for each category.

- Different policies
  - Inter-AS: control over how (foreign) traffic is routed via the own network
  - Intra-AS: control over how traffic is routed within the the own network
- Scale
  - Hierarchical routing saves table size, reduced update traffic

Intra- vs. inter-AS routing

- Performance
  - Intra-AS: can focus on performance
  - Inter-AS: policy may dominate over performance
- Example:
  - Intra-AS: RIP, OSPF
  - Inter-AS: BGP

Thank you

Any questions?