CS 4410
Operating Systems

Networking:
Routing Algorithms

Summer 2016

Cornell University

Today

- Dijkstra's algorithm
- Distance-Vector (DV) algorithm
- Hierarchical Routing

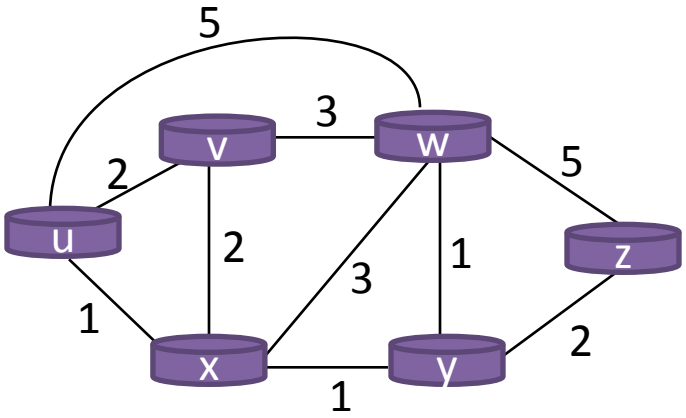
- Resources:
  - http://www-net.cs.umass.edu/kurose-ross-ppt-6e/
  - *Computer Networking: A Top-Down Approach*
    J.F. Kurose and K.W. Ross

The routing problem

- A host is usually attached directly to one router: *default router*.
- *Source router*: default router of the source host.
- *Destination router*: default router of the destination host.
- Target: route a packet from source router to destination router.
  - Given a set of routers connected with links, a routing algorithm finds a “good” path from source router to destination router.
  - “good” is usually “low cost” (e.g., length, speed, money).

Least-cost path

- A graph G is used to formulate routing problems.
- G=(N,E)
  - N: nodes that represent routers
  - E: edges that represent physical links
- Each edge has a value representing its cost.
- Find a path between the source and destination that has least cost.

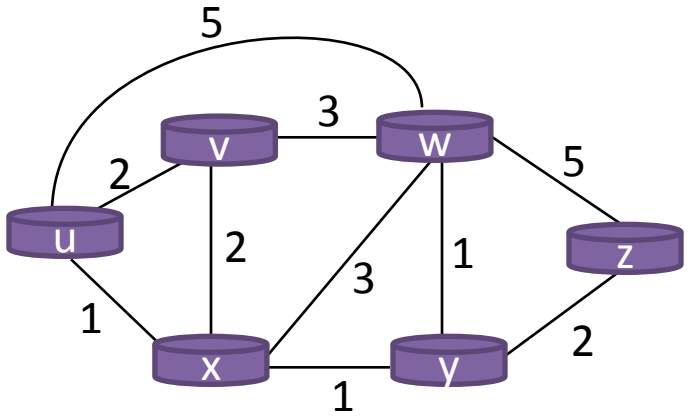


Dijkstra's algorithm

- Compute the least-cost path from one node to all other nodes in the network.
- Iterative algorithm.
  - After the kth iteration, the least-cost paths for k destination nodes are found.
- D(v): cost of the least-cost path from source node to destination v
- p(v): previous node of v along the least-cost path from source.
- N': set of nodes to which the least-cost path is found.

Dijkstra's algorithm: Example

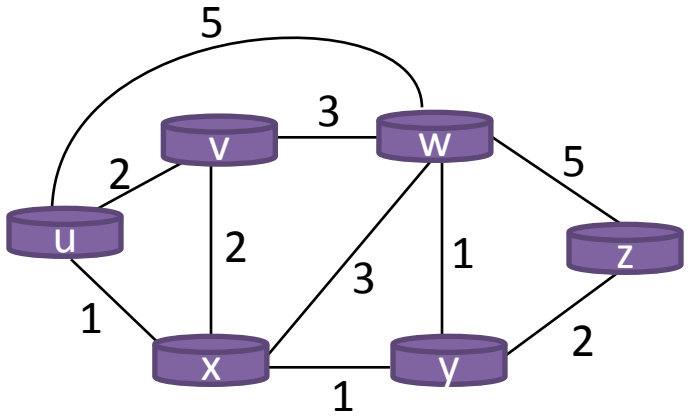
- Source is node u.



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

Dijkstra's algorithm: Example

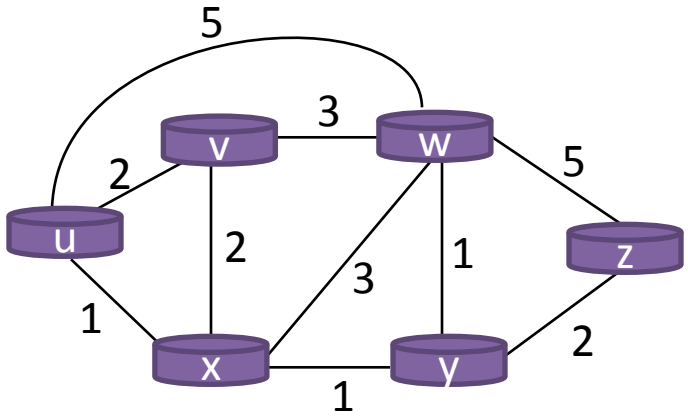
- Source is node u.



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

Dijkstra's algorithm: Example

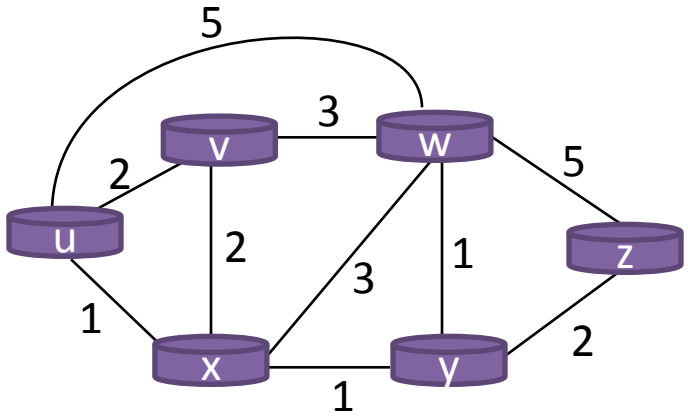
- Source is node u.



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

Dijkstra's algorithm: Example

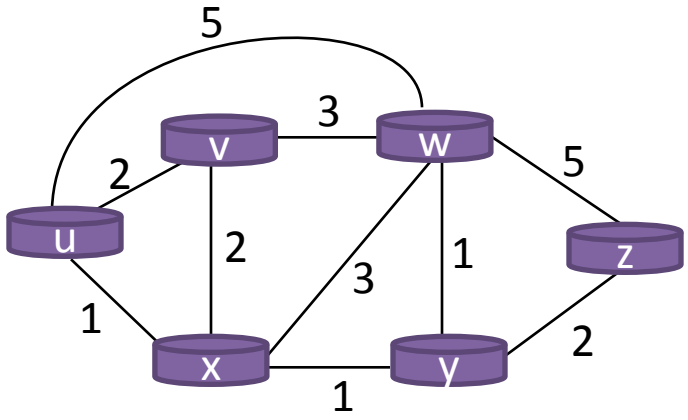
- Source is node u.



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

Dijkstra's algorithm: Example

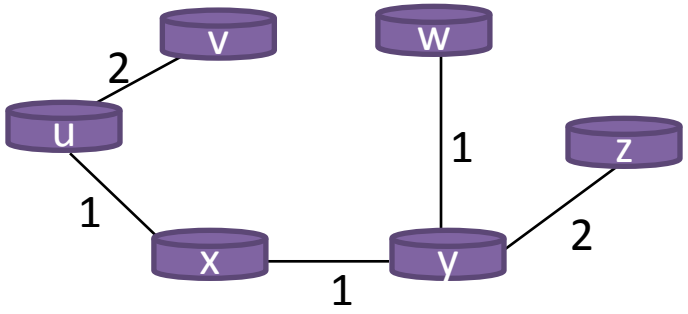
- Source is node u.



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

Dijkstra's algorithm: Example

Resulting shortest-path tree from u:



Resulting forwarding table in u:

Destination	Link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Dijkstra's algorithm

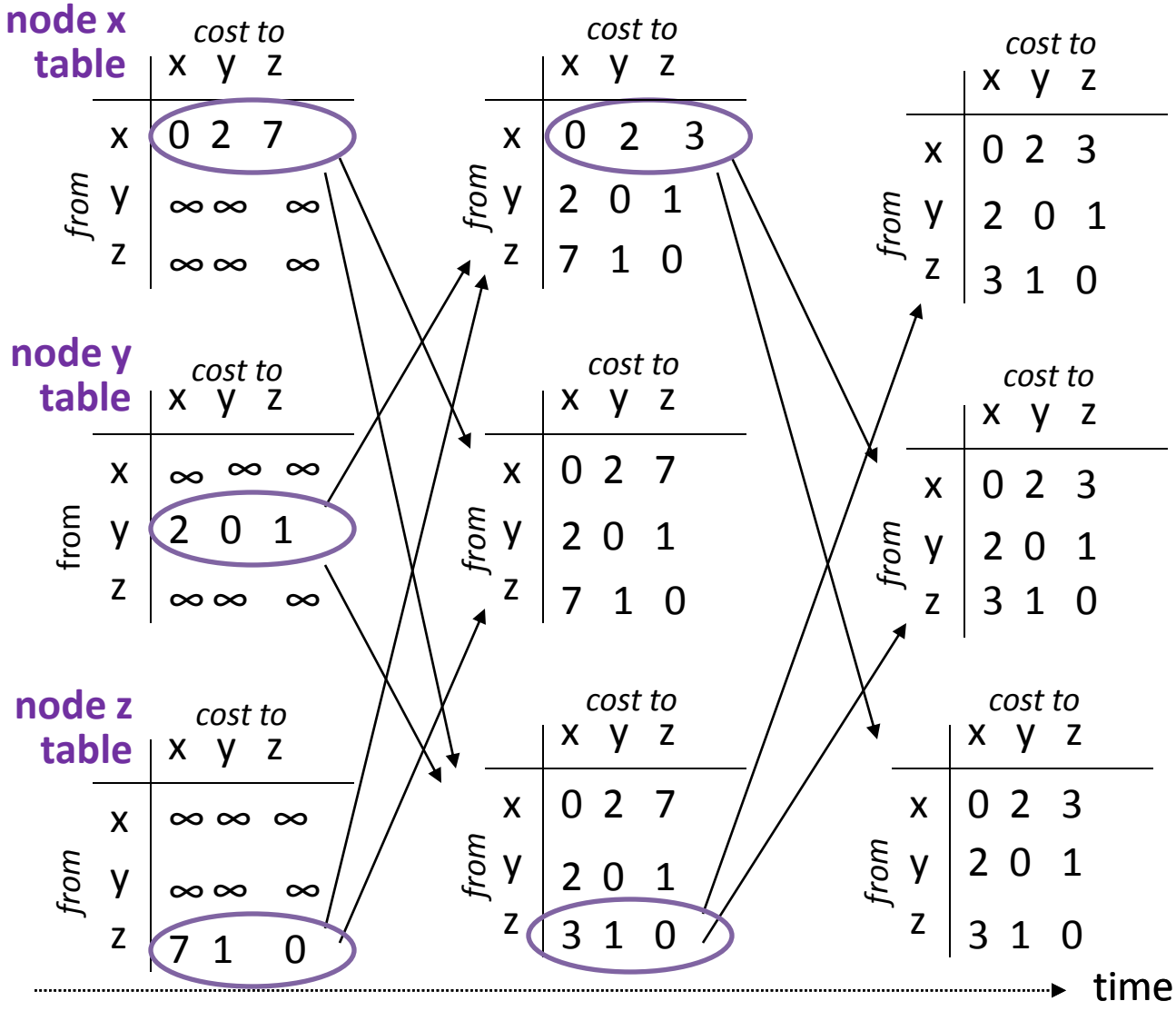
- *Global* routing algorithm:
  - It takes the connectivity between all nodes and all link costs as inputs.
  - Source u needs to have global knowledge of the network in order to determine its forwarding table.

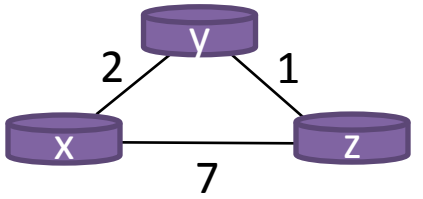
Distance-Vector (DV) algorithm

- *Decentralized* algorithm:
  - No node has complete information about the costs of all links.
  - Each node begins with only the knowledge of the costs of its own directly attached links.
  - Then, each node gradually calculates the least-cost path to a destination by exchanging information with its neighboring nodes.

DV algorithm

- Each node x begins with an estimate $D_x(y)$ of the cost of the least-cost path from itself to y, for all nodes.
  - *Distance vector* of x: $D_x = [D_x(y): y \in N]$
- Node x knows the cost c(x,v) for each neighbor v.
- Neighbors exchange their distance vectors.
- When x receives v's distance vector, it uses *Bellman-Ford* equation to update its own distance vector:
  - $D_x(y) = \min_v\{c(x,v) + D_v(y)\}$ for each node $y \in N$
- If x's distance vector changed, x sends its distance vector to its neighbors.
- If nodes continue exchanging updated distance vectors, each cost estimate $D_x(y)$ will converge to the actual least-cost from x to y.


node x table
cost to
x y z
from
x 0 2 7
y ∞ ∞ ∞
z ∞ ∞ ∞
node y table
cost to
x y z
from
x ∞ ∞ ∞
y 2 0 1
z ∞ ∞ ∞
node z table
cost to
x y z
from
x ∞ ∞ ∞
y ∞ ∞ ∞
z 7 1 0
cost to
x y z
from
x 0 2 3
y 2 0 1
z 7 1 0
cost to
x y z
from
x 0 2 7
y 2 0 1
z 7 1 0
cost to
x y z
from
x 0 2 7
y 2 0 1
z 3 1 0
cost to
x y z
from
x 0 2 3
y 2 0 1
z 3 1 0
cost to
x y z
from
x 0 2 3
y 2 0 1
z 3 1 0
cost to
x y z
from
x 0 2 3
y 2 0 1
z 3 1 0
time

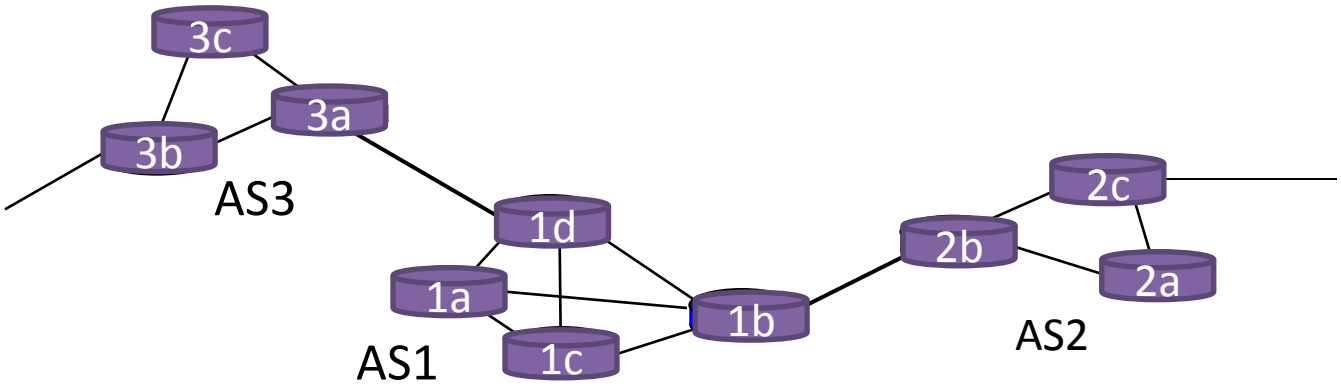

y
2
1
x
z
7

Hierarchical Routing

- As the number of routers become large, the overhead involved in maintaining routing information becomes prohibitive.
- Internet providers want to manage their network as they wish, while still being able to connect to other networks.
- Organizing routers into *autonomous systems* (ASs) solve these problems.

Hierarchical Routing

- Routers within the same AS all run the same routing algorithm (e.g., Dijkstra or DV).
  - *Intra-AS* routing protocol
- One or more routers in an AS are responsible to forward packets to destinations outside AS.
  - *Gateway* routers



Hierarchical Routing

- How to route packets outside an AS?
- *Inter-AS* routing protocol:
  - Obtain reachability information from neighboring ASs, and
  - Propagate the reachability information to all routers in AS.
- In the Internet, all ASs run the same inter-AS routing protocol: *BGP (*Border Gateway Protocol*)*
  - Uses a DV-like algorithm.

Today

- Dijkstra's algorithm
- Distance-Vector (DV) algorithm
- Hierarchical Routing

- Resources:
  - http://www-net.cs.umass.edu/kurose-ross-ppt-6e/
  - *Computer Networking: A Top-Down Approach*
    J.F. Kurose and K.W. Ross

Coming up…

- Next lecture: Transport layer
- HW5:
  - Released today
  - Due on Wednesday