CN-Advanced
L31

Link State Routing

Dr. Ram P Rustagi
rprustagi@ksit.edu.in
http://www.rprustagi.com
https://www.youtube.com/rprustagi

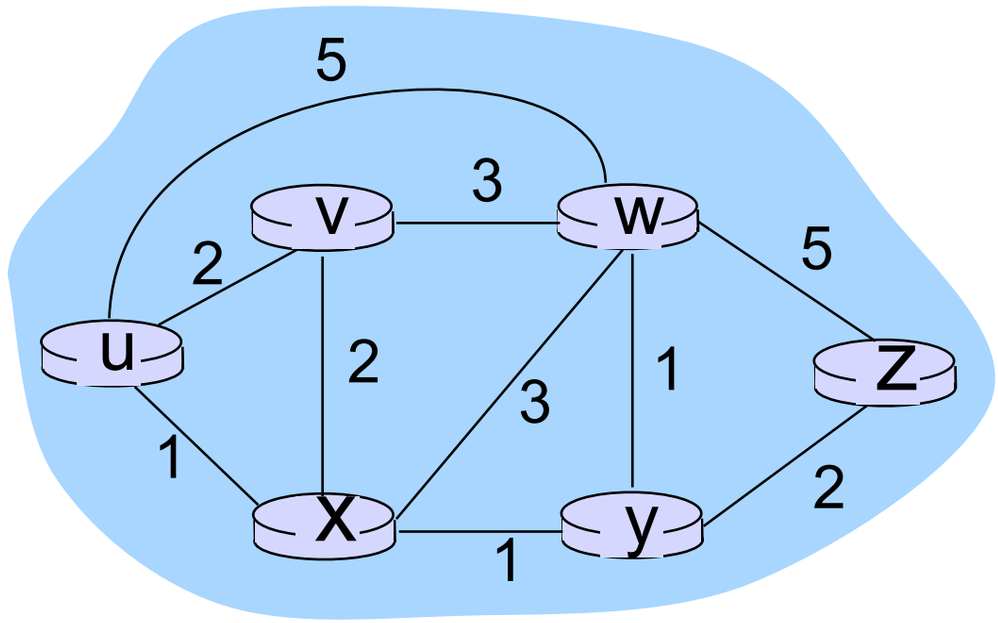
Routing Algorithm

- Purpose:
  - Given set of routers connected via links
    - find a good path from source to destination
- What is good path
  - Least cost
    - How to define the cost
      - Link length, links speed, monetary cost, others
  - Policy compliance
    - Organisation Y may not forward packet which originated from organisation X

Network as a graph

- Routing is essentially graph theory problem
- A graph represents the network
- Routers are nodes and links are edges
  - Each edge has an associated cost
- Basic problem: find least cost path between two nodes
  - Sum of costs of edges that make up the path
- Is it one time computation?
  - Nodes/links goes up/down
  - Link cost changes
  - New nodes get added (old gets deleted)
- Essentially, network topology keeps on changing
- Hence need **routing protocols**
- **Convergence:** getting consistent routing information

Graph abstraction



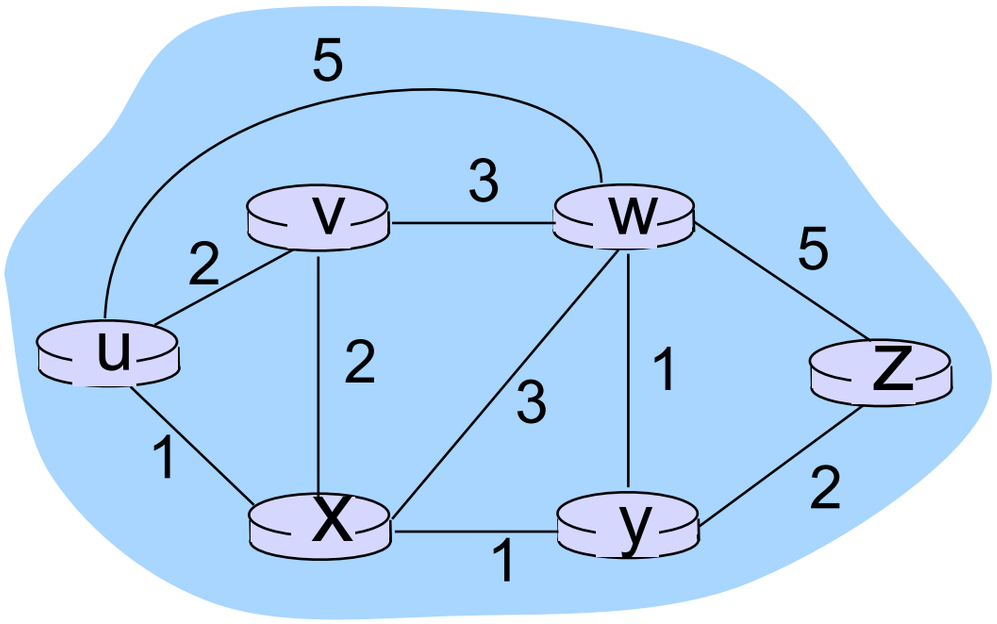
graph: G = (N,E)

N = set of routers = { u, v, w, x, y, z }

E = set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

aside: graph abstraction is useful in other network contexts, e.g., P2P, where *N* is set of peers and *E* is set of TCP connections

Graph abstraction: costs



c(x,x') = cost of link (x,x')
e.g., c(w,z) = 5

cost could always be 1, or inversely related to bandwidth, or inversely related to congestion

cost of path $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

key question: what is the least-cost path between u and z ?
How many paths you considered?
if all edges have same cost, then it is shortest path

routing algorithm: algorithm that finds that least cost path

Routing algorithm classification

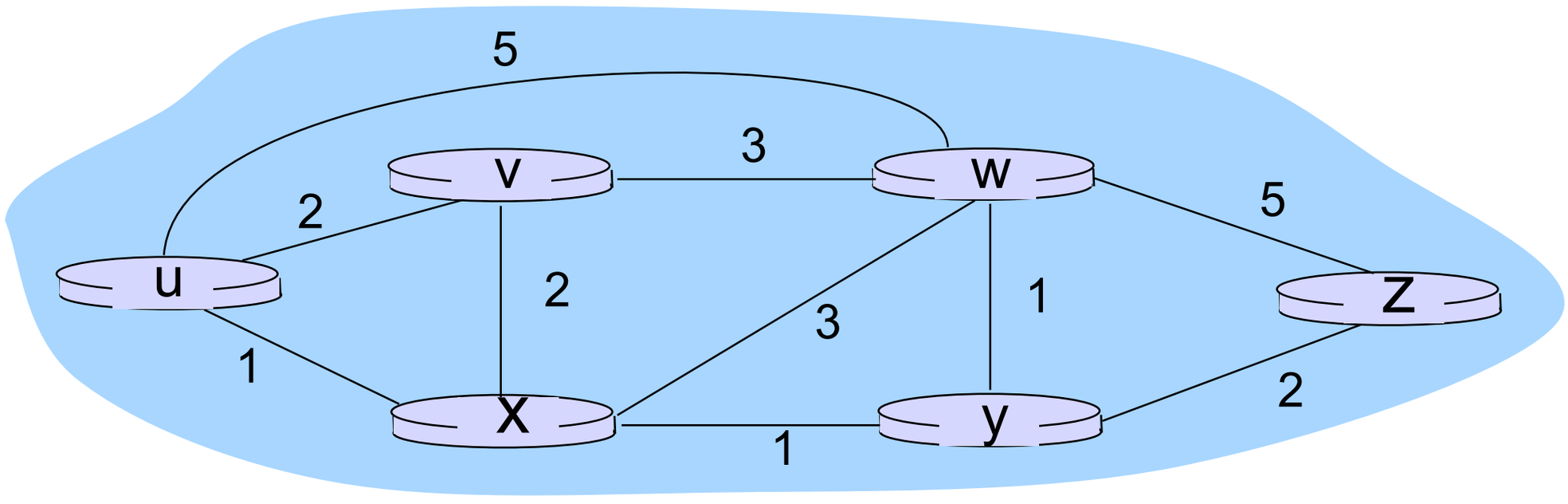
- *Q: Global or decentralized information?*
- *Global:*
- All routers have complete topology, link cost info
- "link state" algorithms
- *Decentralized:*
- Router knows physically-connected neighbors, link costs to neighbors
- Iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

- *Q: Static or dynamic?*
- *Static:*
- Routes change slowly over time
- *Dynamic:*
- Routes change more quickly
- Periodic update
- When link cost changes

- *Q: Load sensitive or insensitive?*
- *Insensitive:*
- Routing independent of load
- *Dynamic:*
- Routes change as load changes

Building Network Topology

- How does a router gets a complete topology
  - e.g. how node `u` knows about the link `w—z`
- Each node broadcasts its connectivity
  - Adds unique id (seq number) to this msg
- Receiving nodes forwards to all other links
  - only if it has not forwarded this particular message
- Thus, after D (network diameter), each node knows the topology



A link-state routing algorithm

Dijkstra's algorithm

- Network topology, link costs known to all nodes
  - Accomplished via "link state broadcast"
  - All nodes have same info
- Computes least cost paths from one node ("source") to all other nodes
  - Derives *forwarding table* for that node
- Iterative: after k iterations, know least cost path to k destinations.

notation:

- c(x,y): link cost from node x to y;
  - =∞ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor of node v along path from source to v
- N': set of nodes whose least cost path definitively known

Dijsktra's algorithm

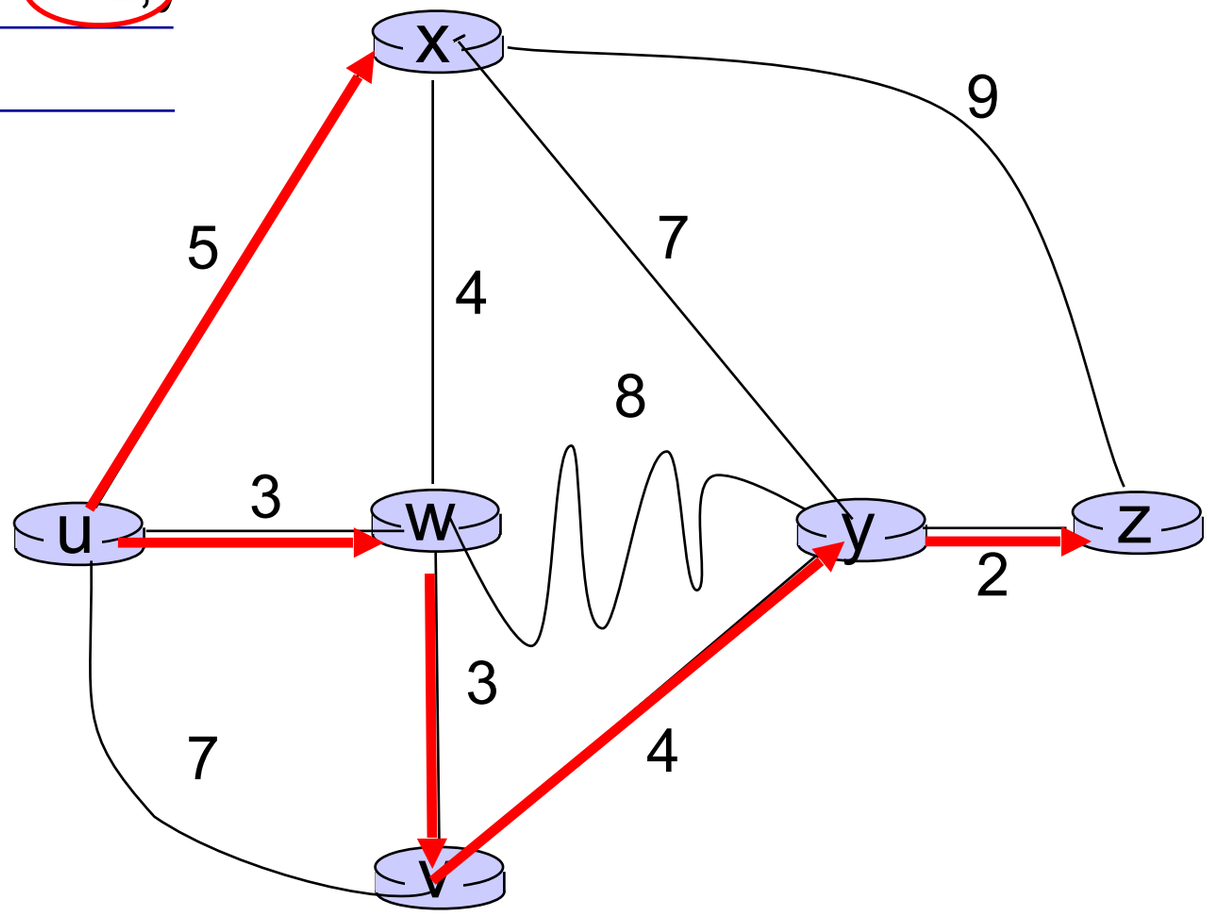
```
1  Initialization:
2    N' = {u}
3    for all nodes v
4      if v adjacent to u
5          then D(v) = c(u,v)
6      else D(v) = ∞
7
8  Loop
9      find w not in N' such that D(w) is a minimum
10     add w to N'
11     update D(v) for all v adjacent to w and not in N' :
12       D(v) = min( D(v), D(w) + c(w,v) )
13     /* new cost to v is either old cost to v or known
14      shortest path cost to w plus cost from w to v */
15 until all nodes in N'
```

Dijkstra's algorithm: example

Step	N'	D(**v**) p(v)	D(**w**) p(w)	D(**x**) p(x)	D(**y**) p(y)	D(**z**) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

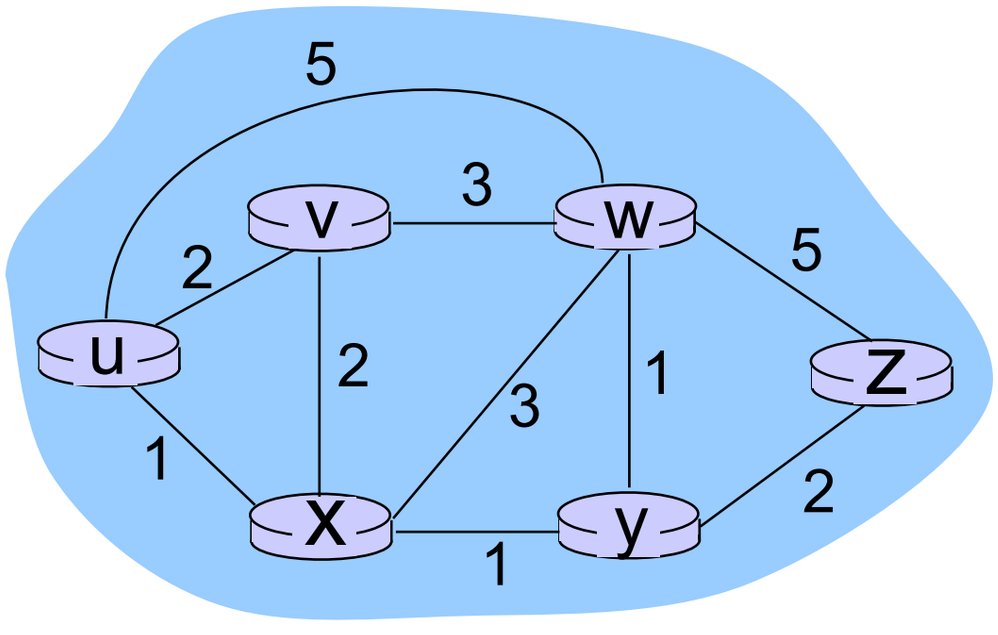
notes:

- construct shortest path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)



Dijkstra's algorithm: another example

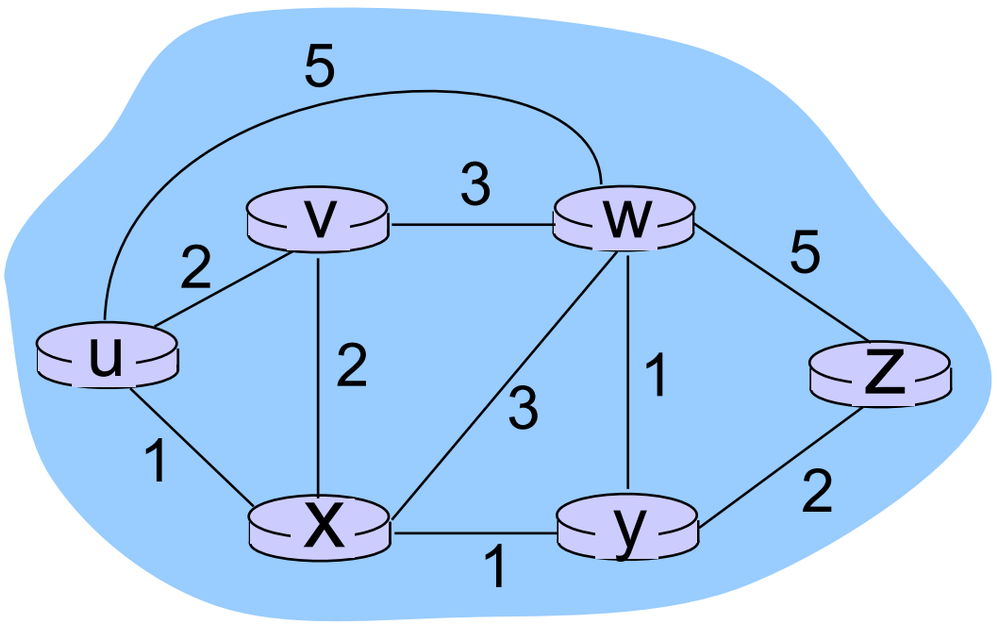
Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

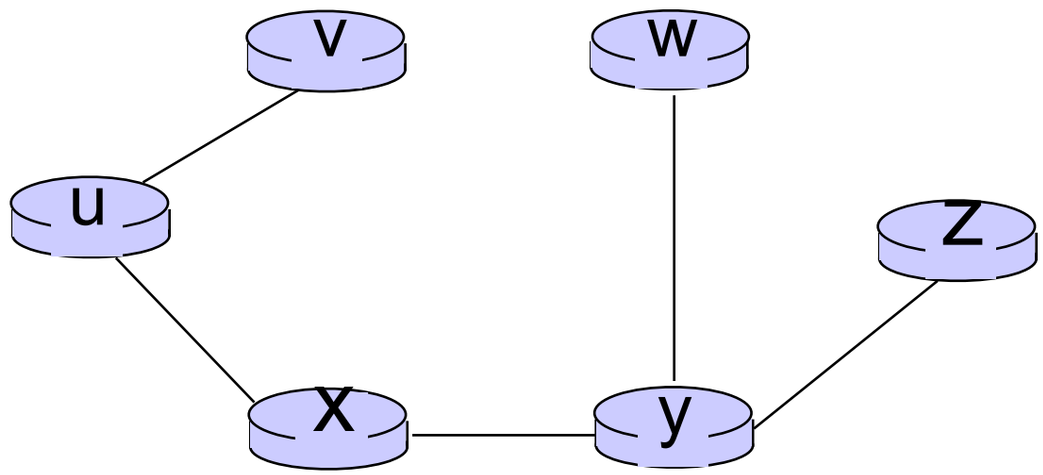


- Check out the online interactive exercises for more examples:
  - http://gaia.cs.umass.edu/kurose_ross/interactive/

Dijkstra's algorithm: example (2)

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)

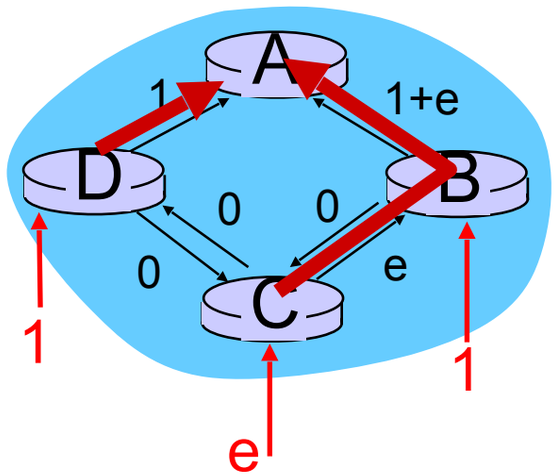
Dijkstra's algorithm, discussion

algorithm complexity: n nodes

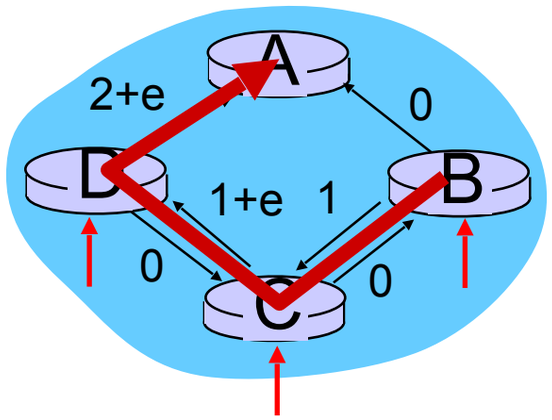
- each iteration: need to check all nodes, w, not in N
- $n(n+1)/2$ comparisons: $O(n^2)$
- more efficient implementations possible: O(mlogn)

oscillations possible:

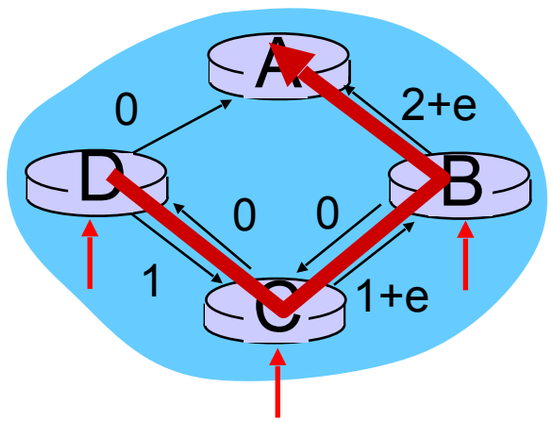
- e.g., support link cost equals amount of carried traffic:



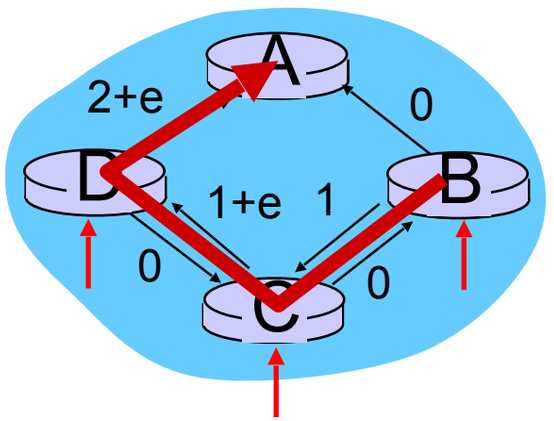
initially



given these costs, find new routing…. resulting in new costs



given these costs, find new routing…. resulting in new costs



given these costs, find new routing…. resulting in new costs

Routing protocols and Metrics

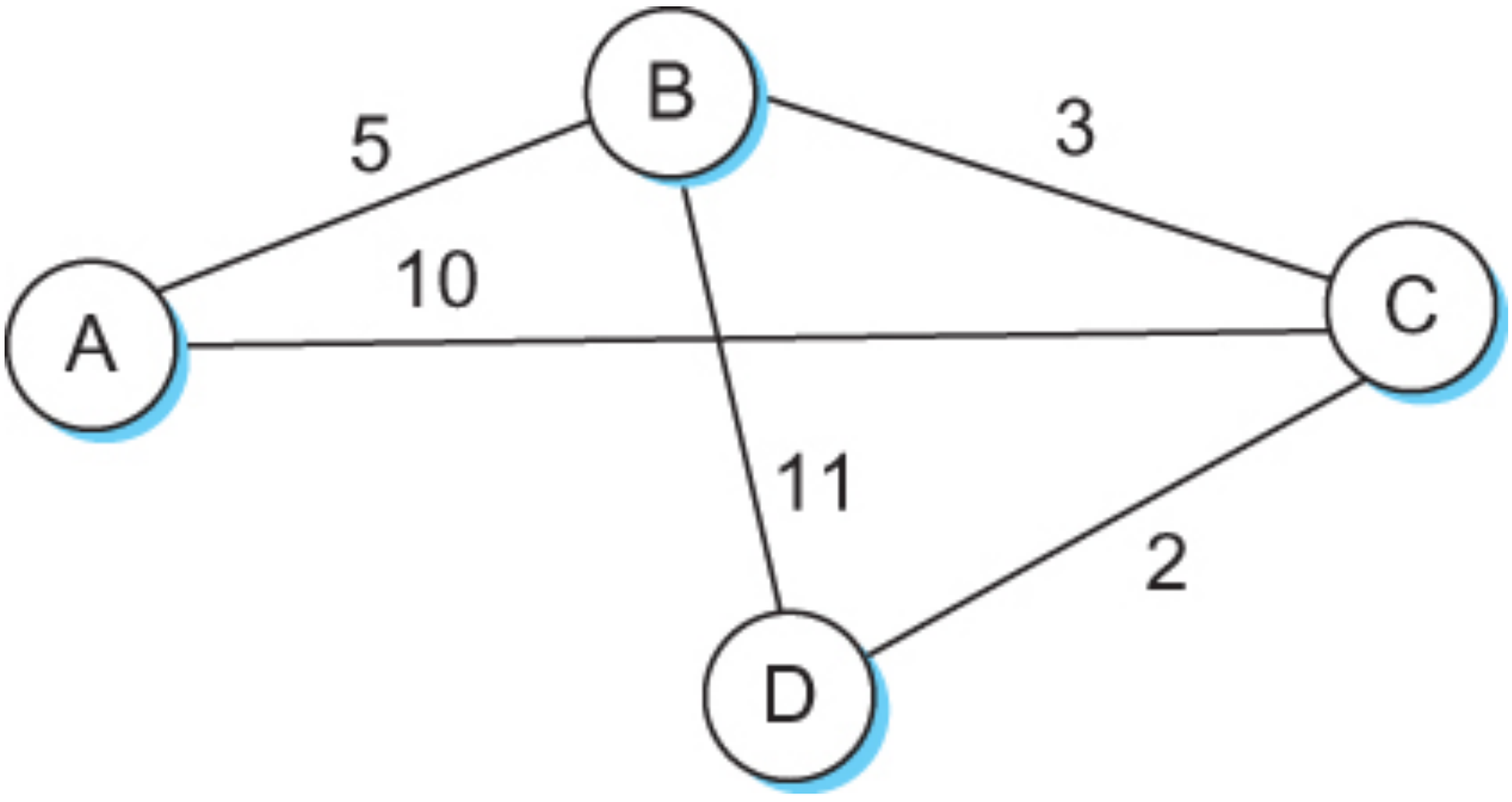
- Hop count
- Bandwidth
- Load - Traffic utilisation
- Delay
- Reliability
- Cost

Linux Routing Table (route -n)

```
Destination     Gateway         Genmask         Flags MSS Window irtt Iface
0.0.0.0         10.1.12.1       0.0.0.0           UG    0   0        0 eth0
0.0.0.0         0.0.0.0         0.0.0.0           U     0   0        0 eth1
10.1.12.0       0.0.0.0         255.255.255.0     U     0   0        0 eth0
10.200.0.0      10.1.13.2       255.255.0.0       UG    0   0        0 eth1
10.250.0.1      0.0.0.0         255.255.255.255   UH    0   0        0 eth1
```

Exercises:

- Find shortest path from D in following network
  - Src: Peterson and Davies, 5th edition



Static vs Dynamic Routing

- **Quiz**:
- Which of the following is applicable to static routing
  - configuration is less error prone
  - more secure because no routing advertisements
  - scaling the network does not pose any problems
  - no computing overhead
  - administrator has less work maintaining the configuration
- Answer
  - more secure
  - no computing overhead

Convergence

- **Q:** Which statement best describes convergence ?
  - Amount of time required for routers to share administrative configuration changes e.g. password
  - The time required for routers in the network to update their routing tables after a topology change has occurred
  - The time required for routers running disparate routing protocols to update their routing tables
- Answer
  - The time required for routers in the network to update their routing tables after a topology change has occurred

Summary

- Network : graph abstraction
- Routing: Shortest path
- Link State: Each node know network topology
- Cost Metric
- Convergence