UNICAST ROUTING PROTOCOLS

A routing table can be either static or dynamic. A static table is one with manual entries.

A dynamic table, on the other hand, is one that is updated automatically when there is a change somewhere in the internet.

Routing protocols have been created in response to the demand for dynamic routing tables.

A routing protocol is a combination of rules and procedures that lets routers in the internet inform each other of changes.

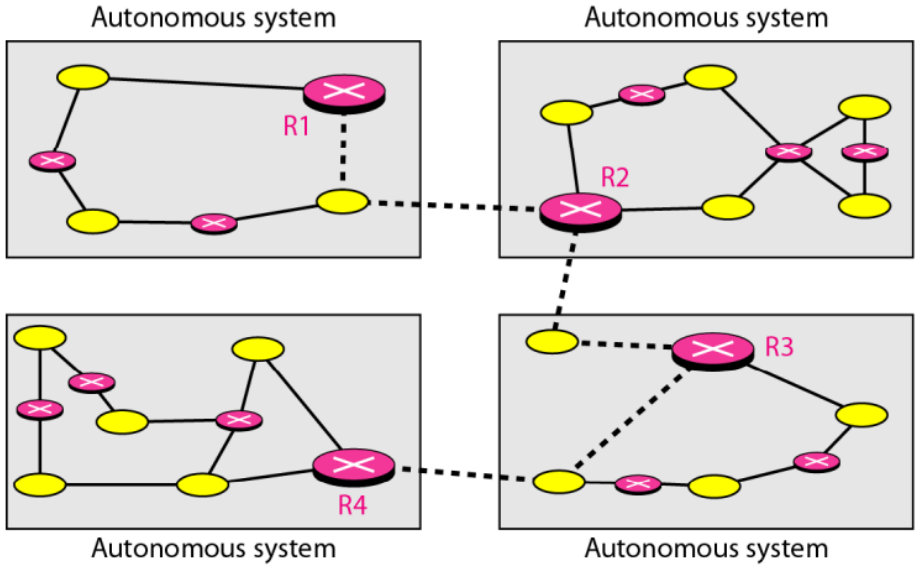
It allows routers to share whatever they know about the internet or their neighborhood.

The routing protocols also include procedures for combining information received from other routers.

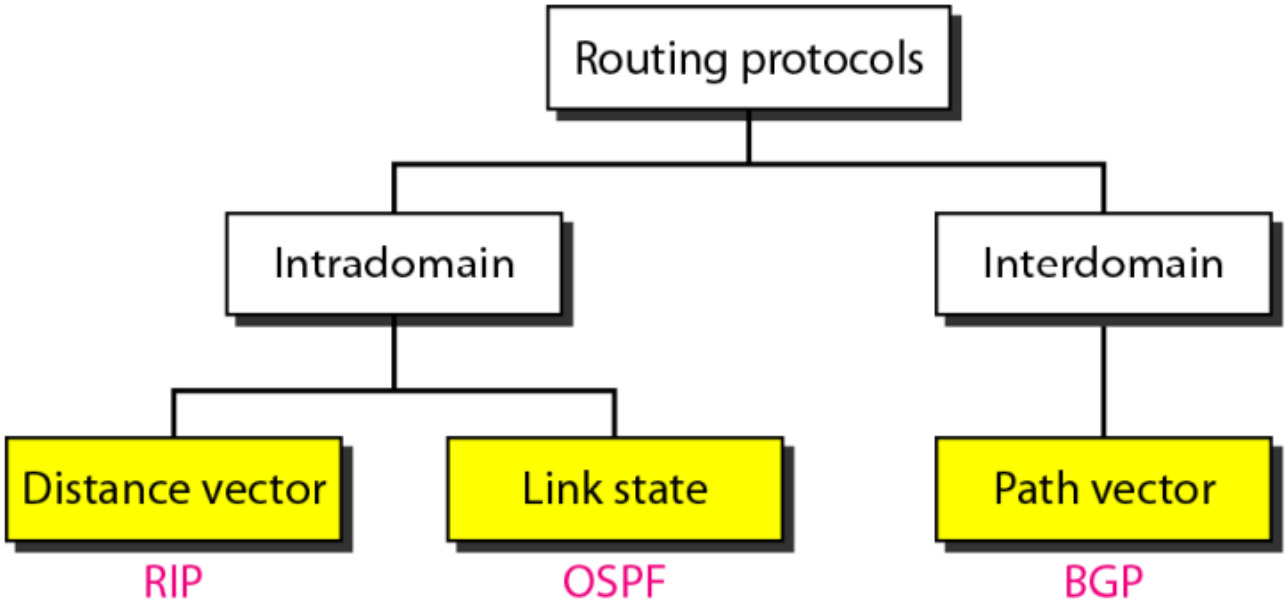
UNICAST ROUTING PROTOCOLS

An autonomous system (AS) is a group of networks and routers under the authority of a single administration. Routing inside an autonomous system is referred to as intradomain routing

Figure 22.12 *Autonomous systems*



UNICAST ROUTING PROTOCOLS



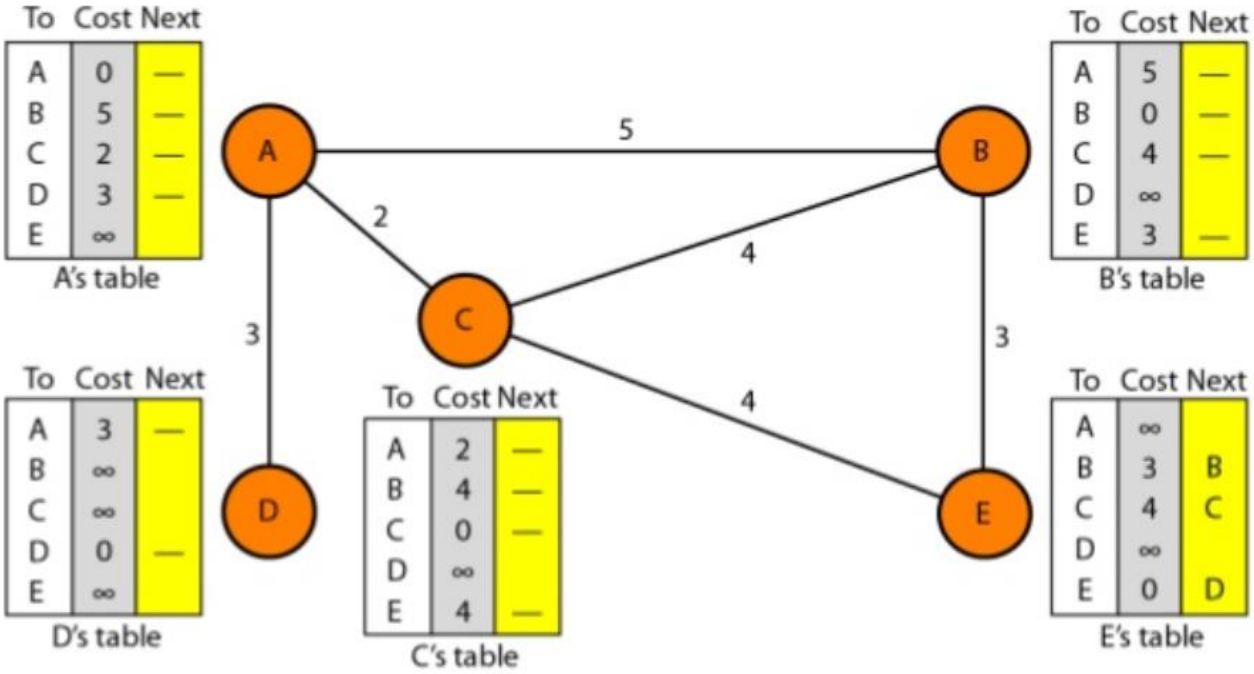
DISTANCE VECTOR ROUTING

A **distance vector routing** algorithm operates by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which link to use to get there.

Each router periodically shares its knowledge about the entire network with its neighbors in following 3 steps:

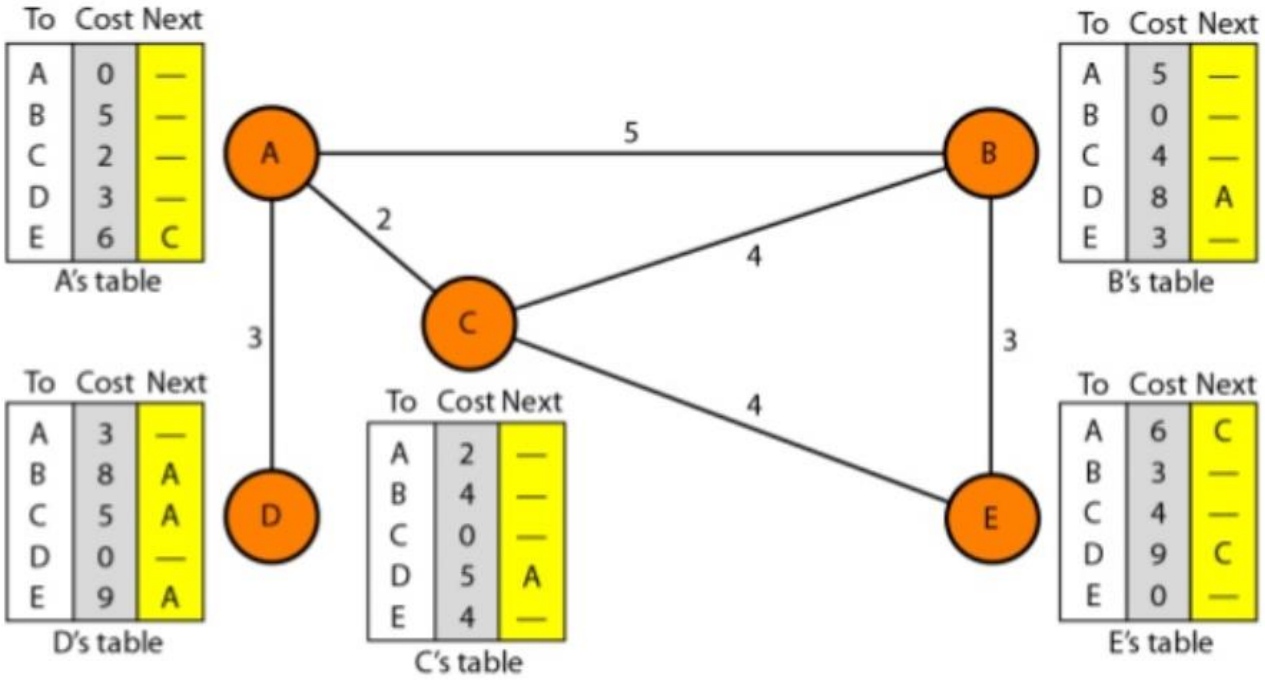
1. Knowledge about the whole network
2. Routing only to neighbors
3. Information sharing at regular intervals

DISTANCE VECTOR ROUTING



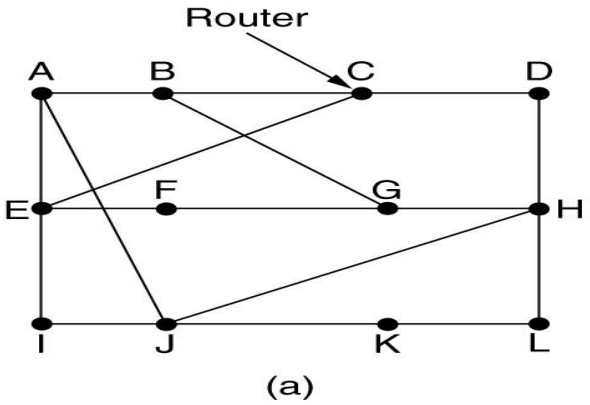
Initialization of tables in DVR

DISTANCE VECTOR ROUTING



Final tables in DVR

3 Distance Vector Routing (1)



(a)

New estimated delay from J

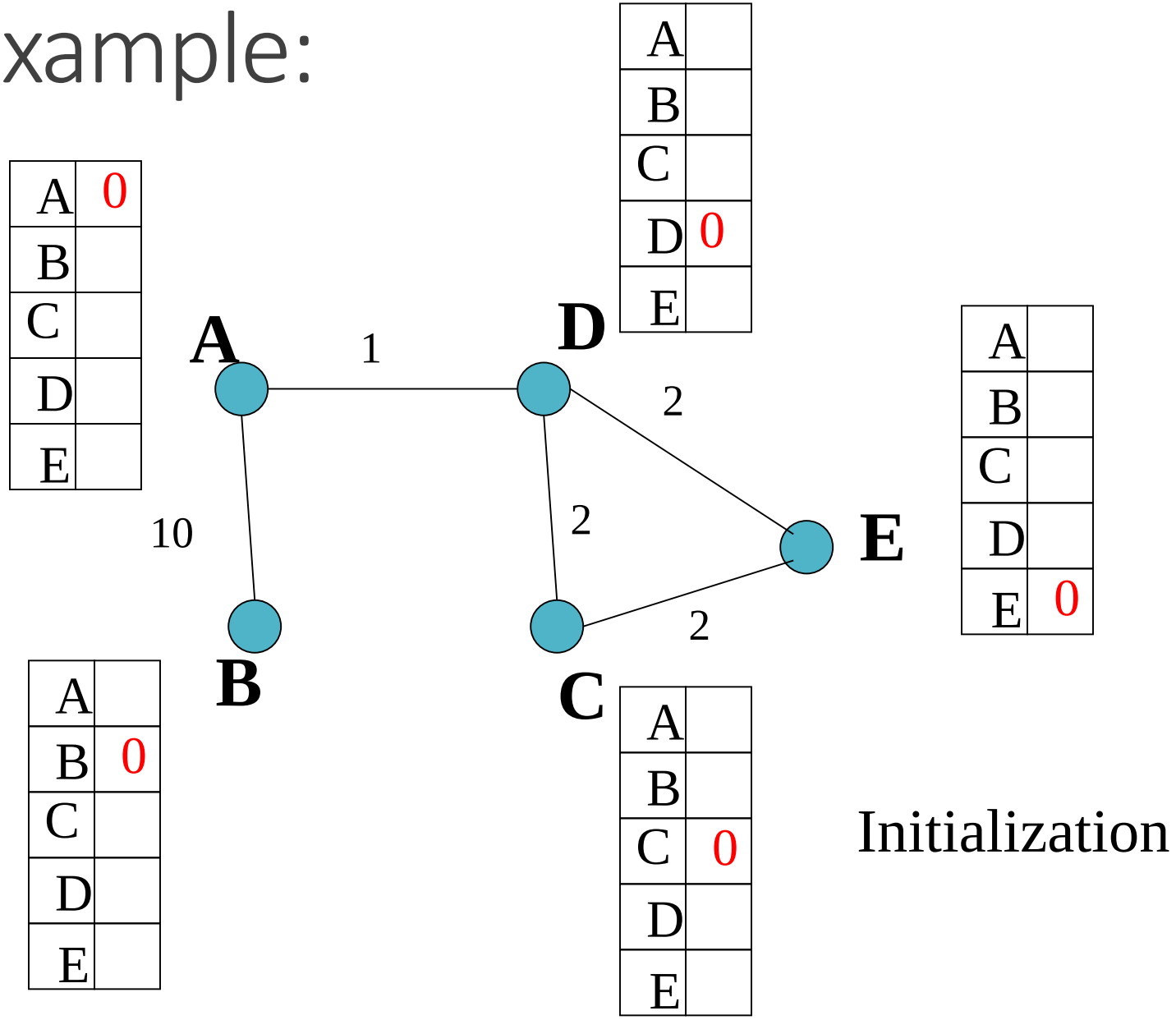
To	A	I	H	K	Line
A	0	24	20	21	8 A
B	12	36	31	28	20 A
C	25	18	19	36	28 I
D	40	27	8	24	20 H
E	14	7	30	22	17 I
F	23	20	19	40	30 I
G	18	31	6	31	18 H
H	17	20	0	19	12 H
I	21	0	14	22	10 I
J	9	11	7	10	0 —
K	24	22	22	0	6 K
L	29	33	9	9	15 K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J

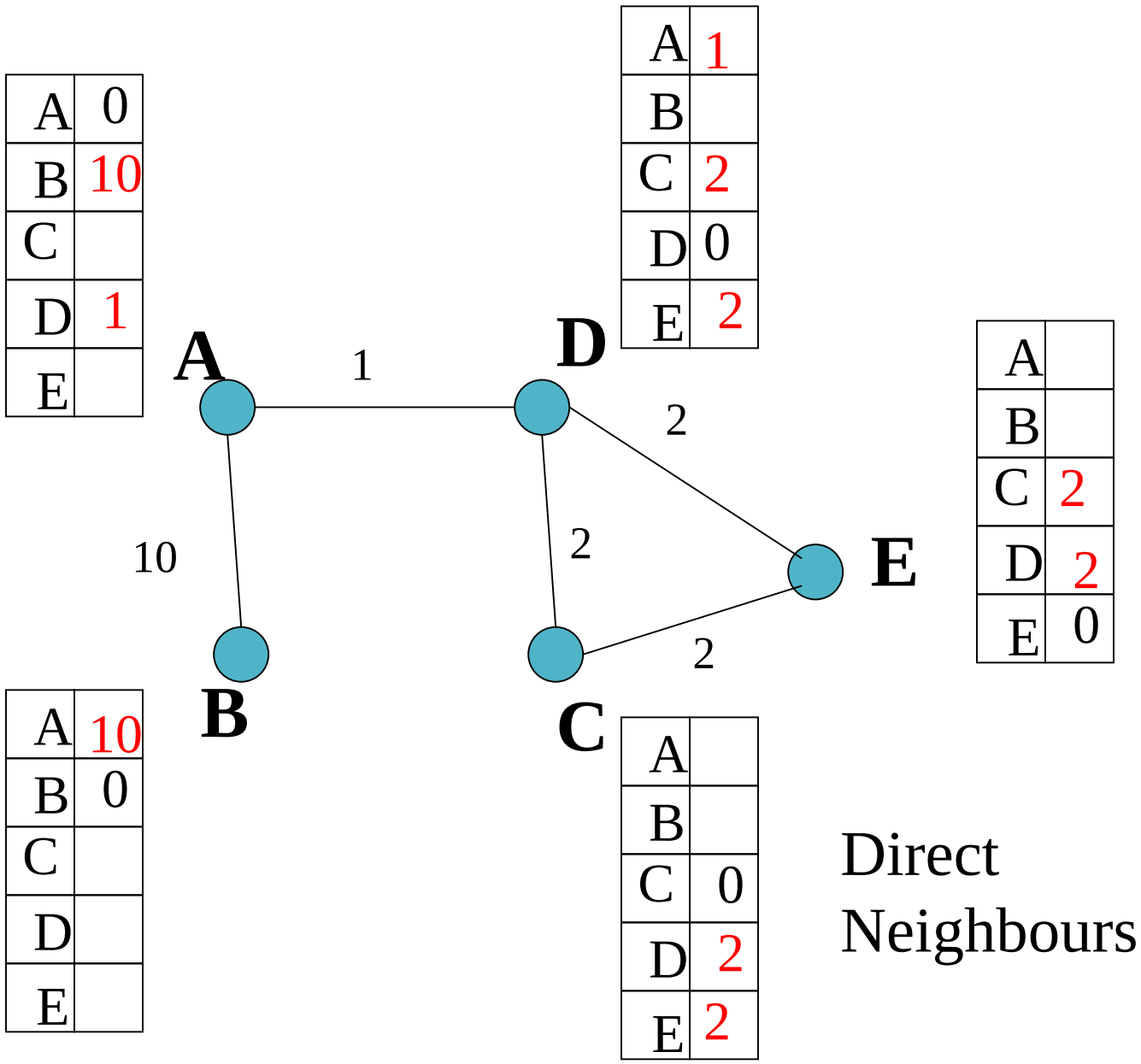
Vectors received from J's four neighbors

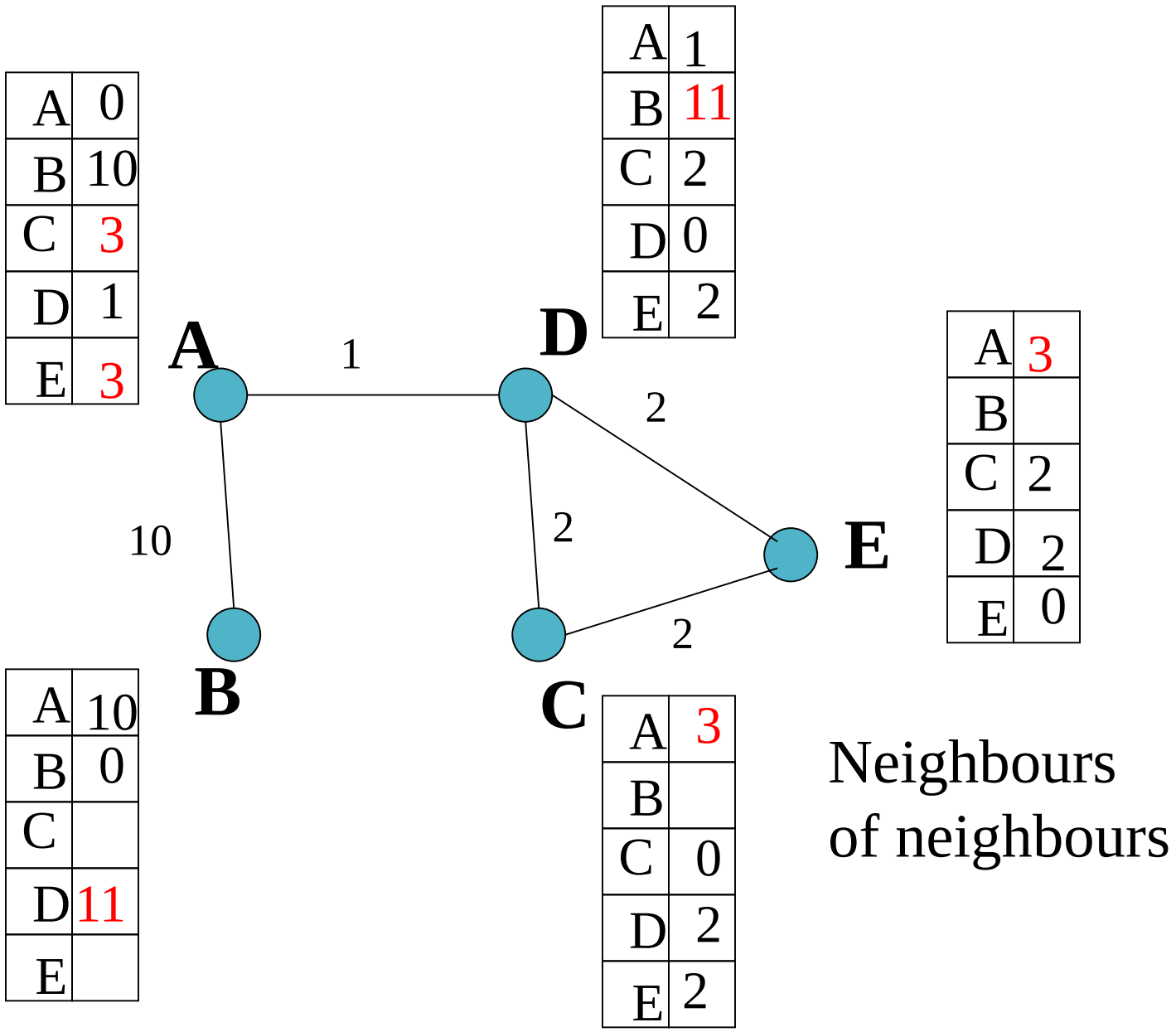
(b)

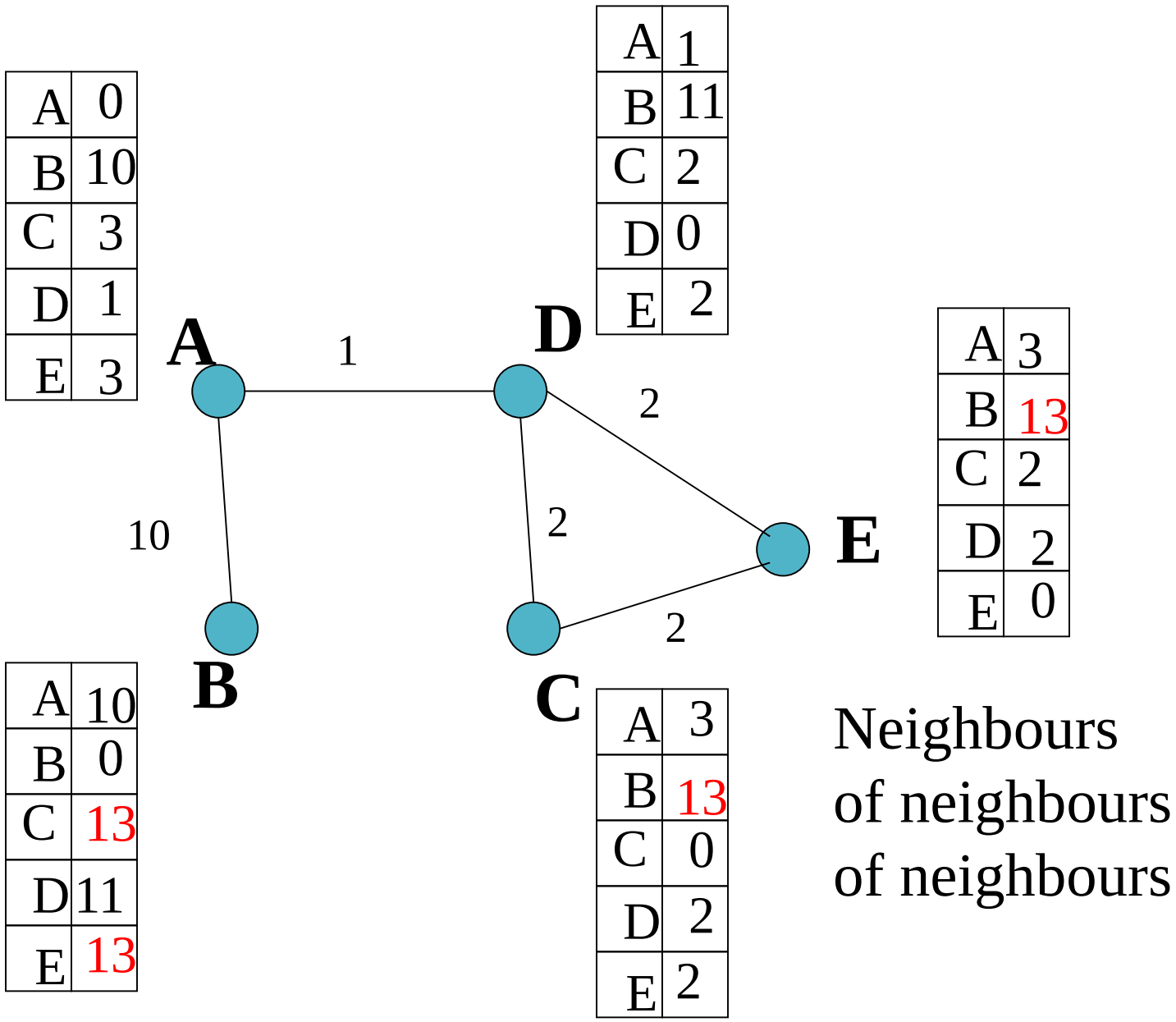
(a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

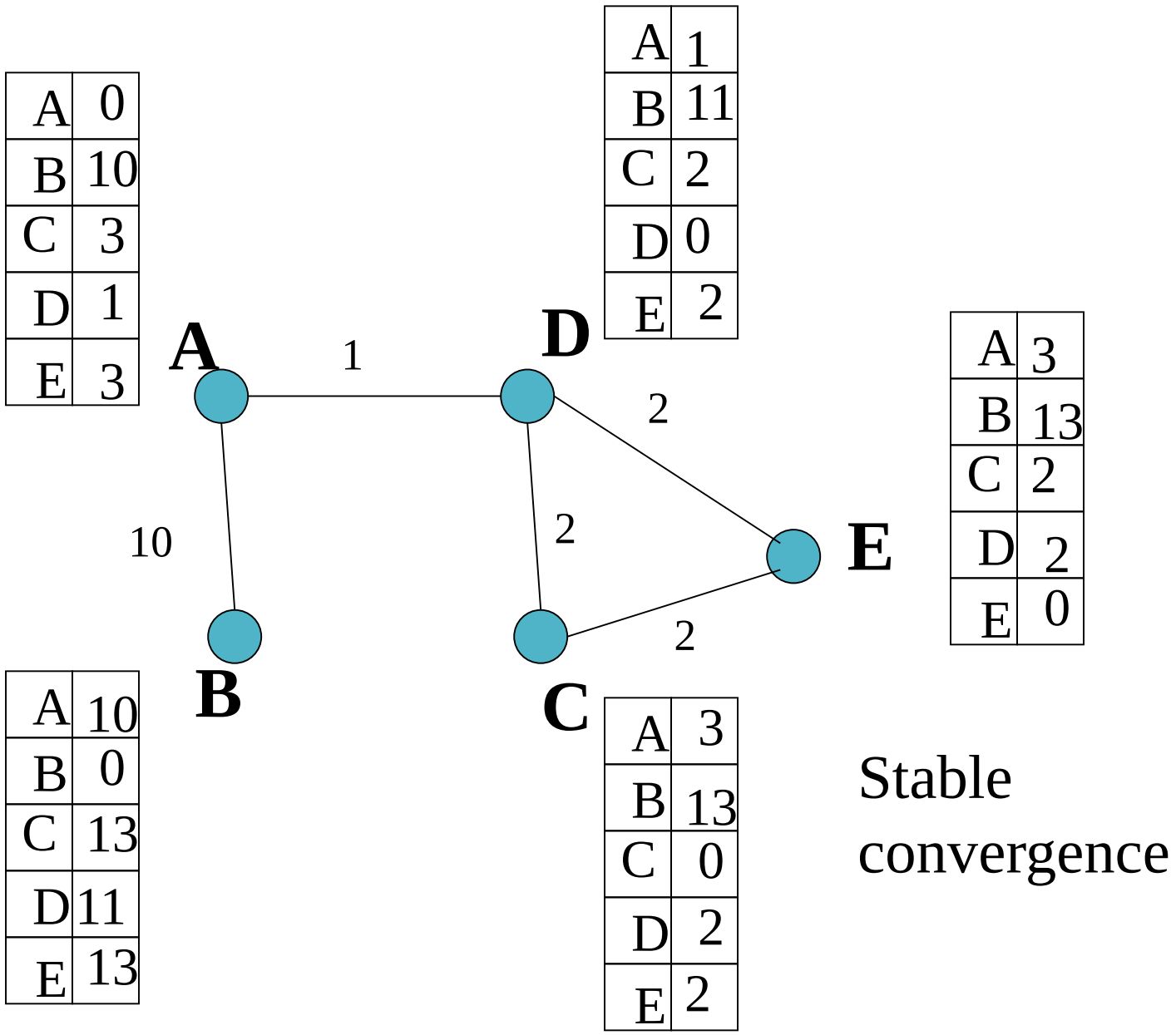
3 Example:

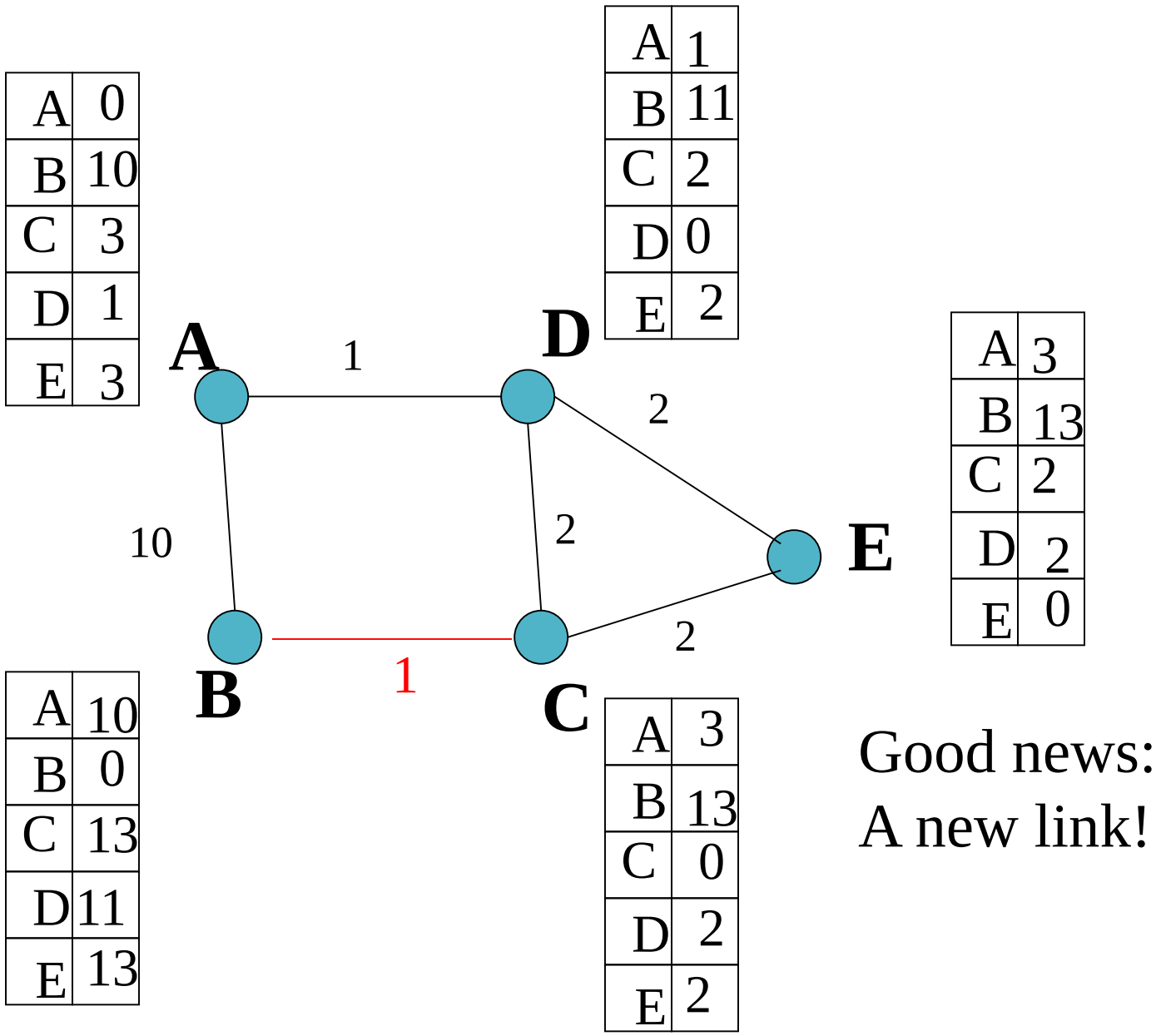

A
B
C
D
E
1
2
2
2
10
A 0
B
C
D
E
A
B
C
D 0
E
A
B
C
D
E 0
A
B 0
C
D
E
A
B
C 0
D
E
Initialization

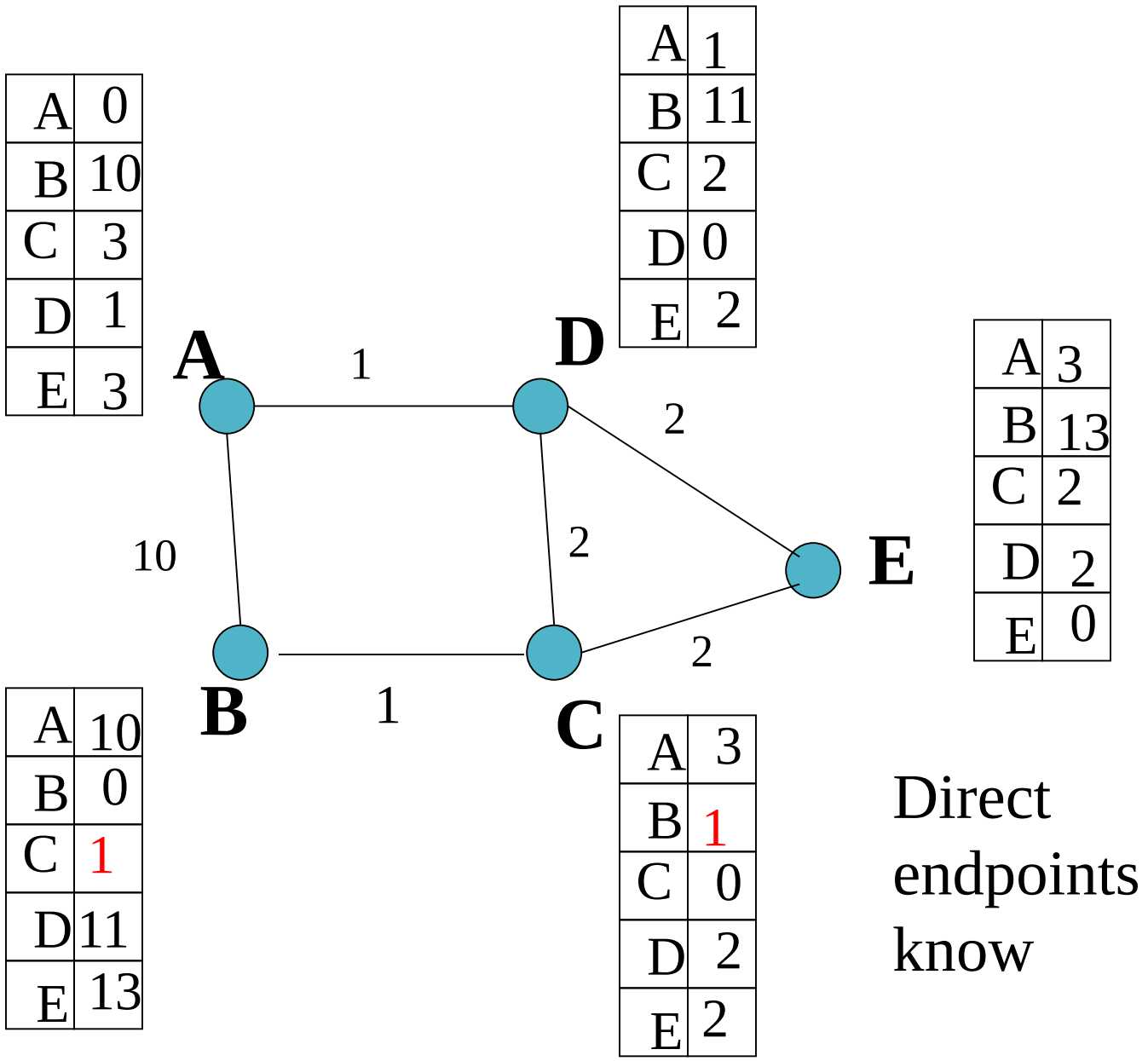

A
B
C
D
E
1
10
2
2
2
A 0
B 10
C
D 1
E
A 1
B
C 2
D 0
E 2
A
B
C 2
D 2
E 0
A 10
B 0
C
D
E
A
B
C 0
D 2
E 2
Direct
Neighbours

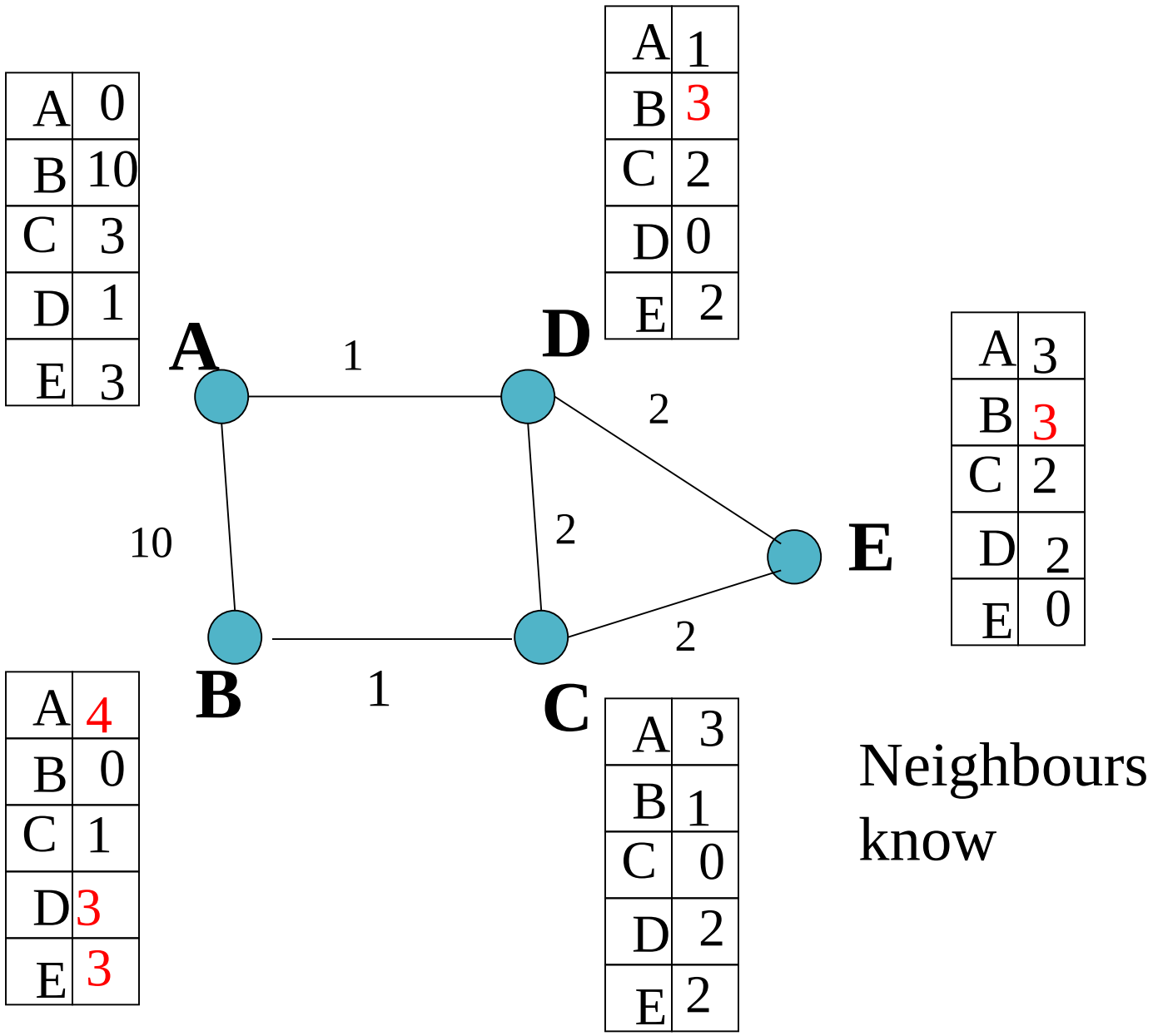

A
B
C
D
E
1
10
2
2
2
A 0
B 10
C 3
D 1
E 3
A 1
B 11
C 2
D 0
E 2
A 3
B
C 2
D 2
E 0
A 10
B 0
C
D 11
E
A 3
B
C 0
D 2
E 2
Neighbours
of neighbours

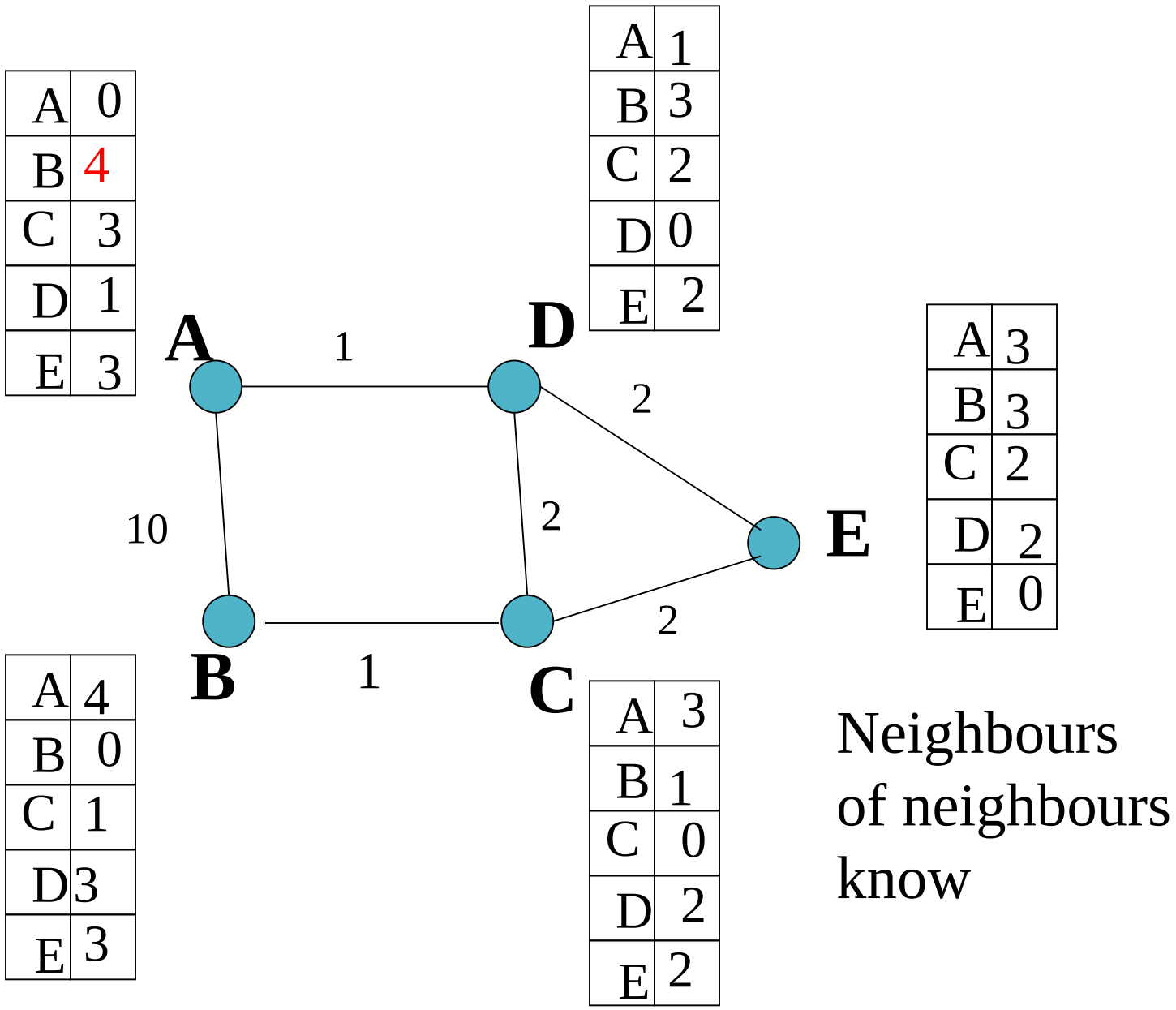

A
B
C
D
E
1
10
2
2
2
A 0
B 10
C 3
D 1
E 3
A 1
B 11
C 2
D 0
E 2
A 3
B 13
C 2
D 2
E 0
A 10
B 0
C 13
D 11
E 13
A 3
B 13
C 0
D 2
E 2
Neighbours
of neighbours
of neighbours

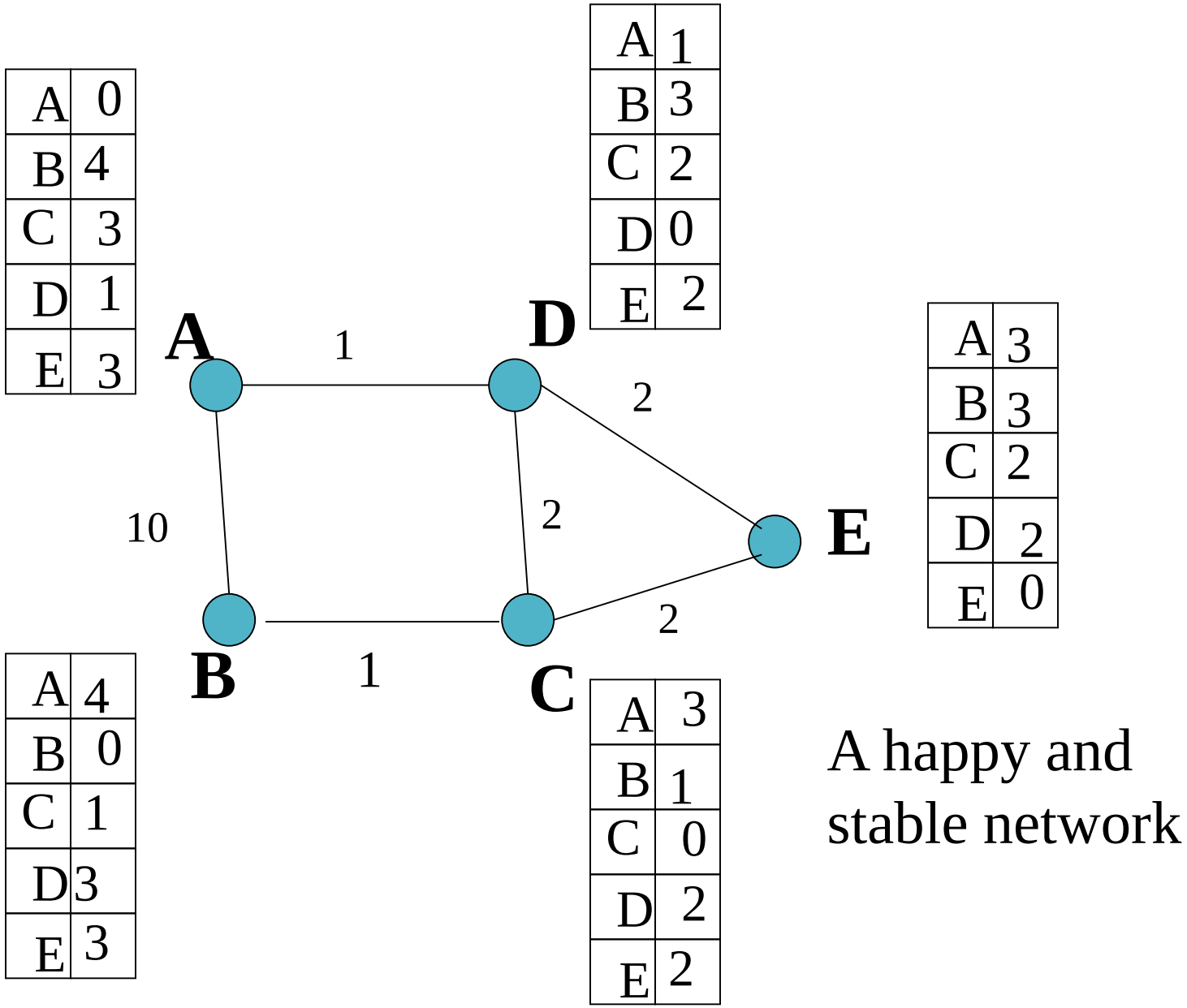

A
B
C
D
E
1
10
2
2
2
A 0
B 10
C 3
D 1
E 3
A 1
B 11
C 2
D 0
E 2
A 3
B 13
C 2
D 2
E 0
A 10
B 0
C 13
D 11
E 13
A 3
B 13
C 0
D 2
E 2
Stable convergence


A
B
C
D
E
1
10
2
2
2
1
A 0
B 10
C 3
D 1
E 3
A 1
B 11
C 2
D 0
E 2
A 3
B 13
C 2
D 2
E 0
A 10
B 0
C 13
D 11
E 13
A 3
B 13
C 0
D 2
E 2
Good news:
A new link!

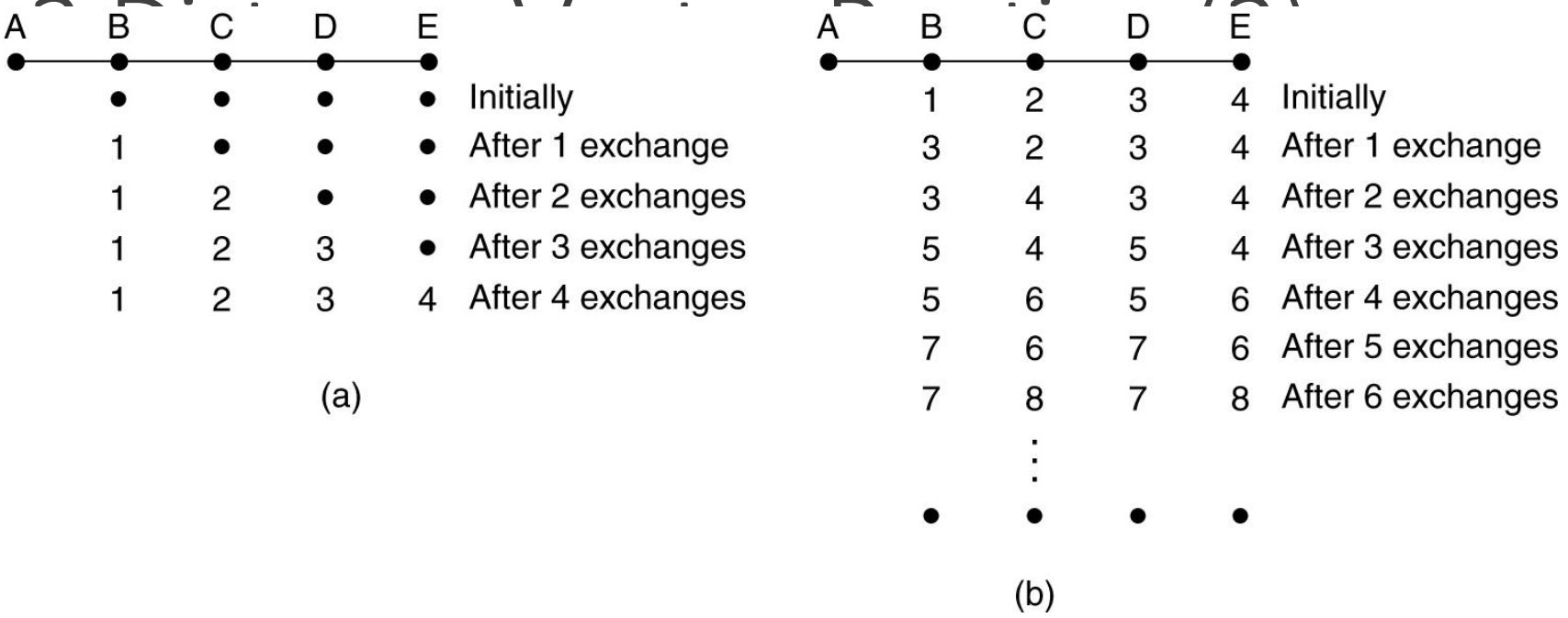

A
D
B
C
E
1
10
2
2
2
1
A 0
B 10
C 3
D 1
E 3
A 1
B 11
C 2
D 0
E 2
A 3
B 13
C 2
D 2
E 0
A 10
B 0
C 1
D 11
E 13
A 3
B 1
C 0
D 2
E 2
Direct endpoints know


A
B
C
D
E
1
10
2
2
2
1
A 0
B 10
C 3
D 1
E 3
A 1
B 3
C 2
D 0
E 2
A 3
B 3
C 2
D 2
E 0
A 4
B 0
C 1
D 3
E 3
A 3
B 1
C 0
D 2
E 2
Neighbours know


A
B
C
D
E
1
10
2
2
2
1
A 0
B 4
C 3
D 1
E 3
A 1
B 3
C 2
D 0
E 2
A 3
B 3
C 2
D 2
E 0
A 4
B 0
C 1
D 3
E 3
A 3
B 1
C 0
D 2
E 2
Neighbours of neighbours know



A happy and stable network



The count-to-infinity problem.

BELLMAN –FORD ALGORITHM

Bellman ford algorithm is a single-source shortest path algorithm. This algorithm is used to find the shortest distance from the single vertex to all the other vertices of a weighted graph.

Bellman ford algorithm guarantees the correct answer even if the weighted graph contains the negative weight values.

BELLMAN –FORD ALGORITHM

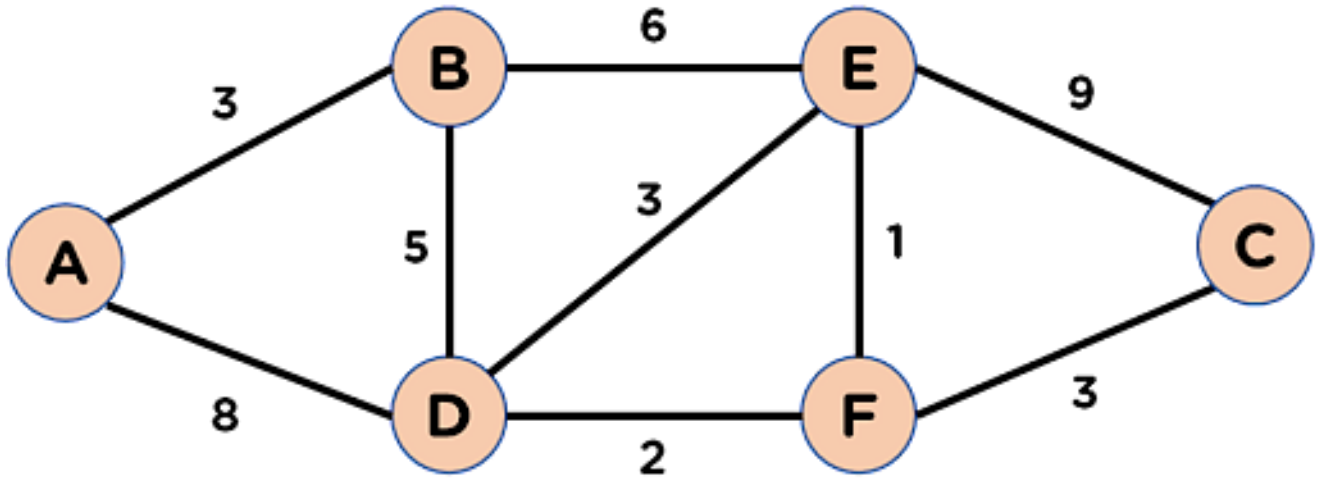
We will go on relaxing all the edges (n - 1) times where,

n = number of vertices

Relaxing means:

$$\text{If } (d(u) + c(u , v) < d(v))$$

$$d(v) = d(u) + c(u , v)$$


A
B
D
E
F
C
3
8
5
6
3
2
1
9
3

DISTANCE VECTOR ROUTING

Updating the Routing table:

When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

1. The receiving node needs to add the cost between itself and the sending node to each value in the second column.
2. The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

DISTANCE VECTOR ROUTING

When to Share

The question now is, When does a node send its partial routing table (only two columns) to all its immediate neighbors? The table is sent both periodically and when there is a change in the table.

Periodic Update : A node sends its routing table, normally every 30 s, in a periodic update.

Triggered Update : A node sends its two-column routing table to its neighbors anytime there is a change in its routing table. This is called a triggered update. The change can result from the following.

1. A node receives a table from a neighbor, resulting in changes in its own table after updating.
2. A node detects some failure in the neighboring links which results in a distance change to infinity.

ROUTING INFORMATION PROTOCOL

Hop Count

A router in an AS needs to know how to forward a packet to different networks in an AS, RIP routers advertise the cost of reaching different networks instead of reaching other nodes**.**

Any route in an AS cannot have more than 15 hops.

RIP Implementation

RIP is implemented as a process that uses the service of UDP.

RIP has two versions: RIP-1 and RIP-2

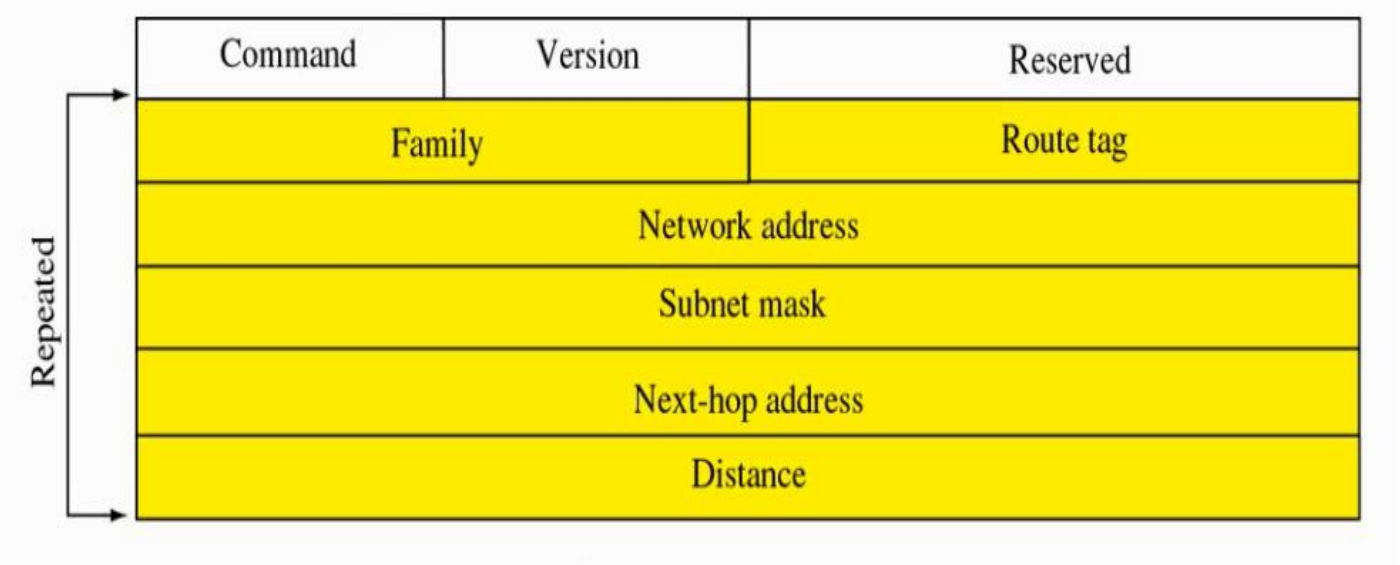
ROUTING INFORMATION PROTOCOL

RIP Messages

RIP has 2 types of messages : Request and Response

A request message can ask about specific entries or all entries.

A response message can be either solicited or unsolicited.



ROUTING INFORMATION PROTOCOL

Algorithm

1. RIP implements the same algorithm as the DVR algorithm with some minor changes.
2. Instead of sending only distance vectors, a router needs to send the whole contents of its forwarding table.
3. The receiver adds one hop to each cost and changes the next router field to the address of the sending router.

Timers in RIP: RIP uses three timers to support its operation.

1. Periodic timer(25-35 seconds)
2. Expiration timer(180 seconds)
3. Garbage Collection timer(120 seconds)

ROUTING INFORMATION PROTOCOL

Version 1 of RIP uses broadcasting to send RIP message to every neighbor .

All the router and the hosts receive the packets .

RIP version 2 Uses the multicast address 224.0.0.9 to multicast RIP message only to RIP routers in the network.

RIP message are encapsulated in UDP user datagram.

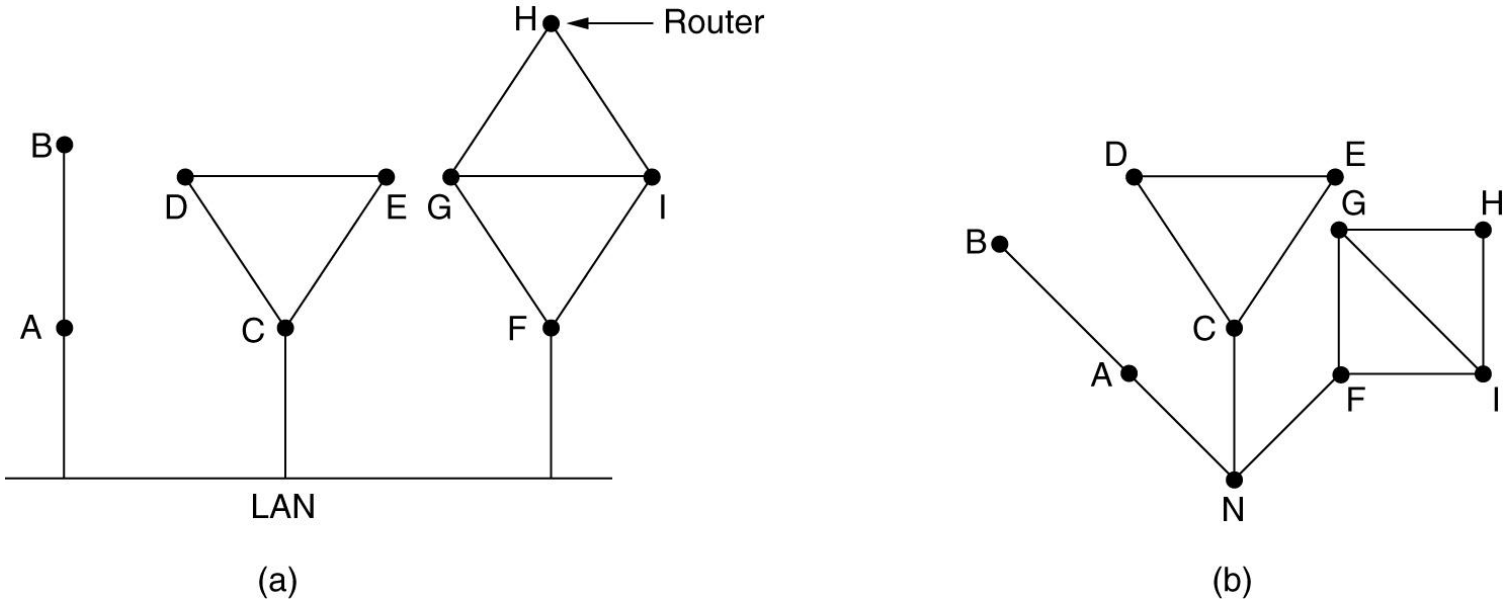
The well-known port assigned to RIP in UDP is port 520.

4 Link State Routing (1)

Each router must do the following:

- Discover its neighbors, learn their network address.
- Measure the delay or cost to each of its neighbors.
- Construct a packet telling all it has just learned.
- Send this packet to all other routers.
- Compute the shortest path to every other router.

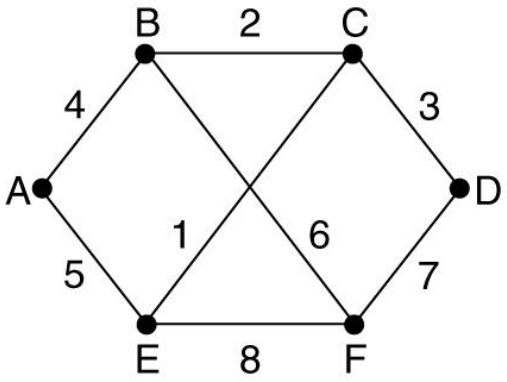
4 Link State Routing (2): Learning about the Neighbors



(a) Nine routers and a LAN.

(b) A graph model of (a).

4 Link State Routing (4): Building Link State Packets



(a)

Link State Packets

A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
E	1

D	
Seq.	
Age	
C	3
F	7

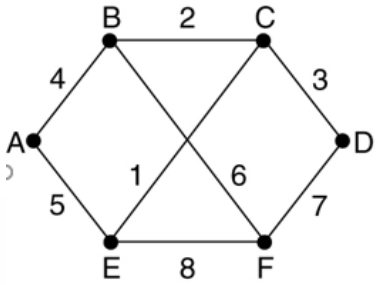
E	
Seq.	
Age	
A	5
C	1
F	8

F	
Seq.	
Age	
B	6
D	7
E	8

(b)

(a) A subnet. (b) The link state packets for this subnet.

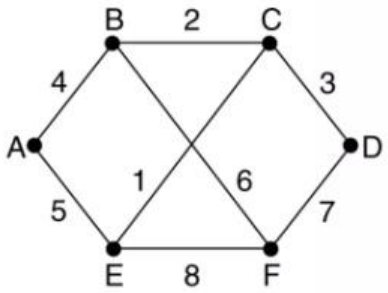
Distributing the Link State Packets



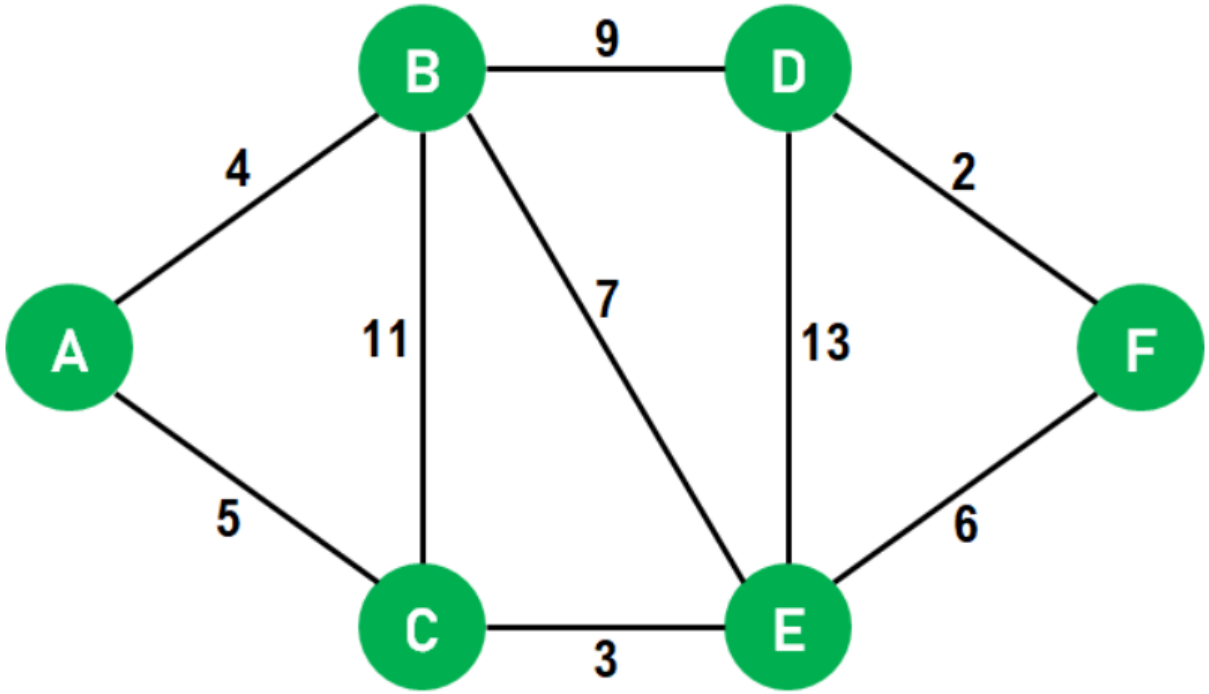
Source	Seq.	Age	Send flags A	Send flags C	Send flags F	ACK flags A	ACK flags C	ACK flags F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

The packet buffer for router B in the previous slide (Fig. 5-13).

Compute Shortest Path – Dijkstra's Algorithm



LINK STATE ROUTING- DIJKSTRAS ALGORITHM



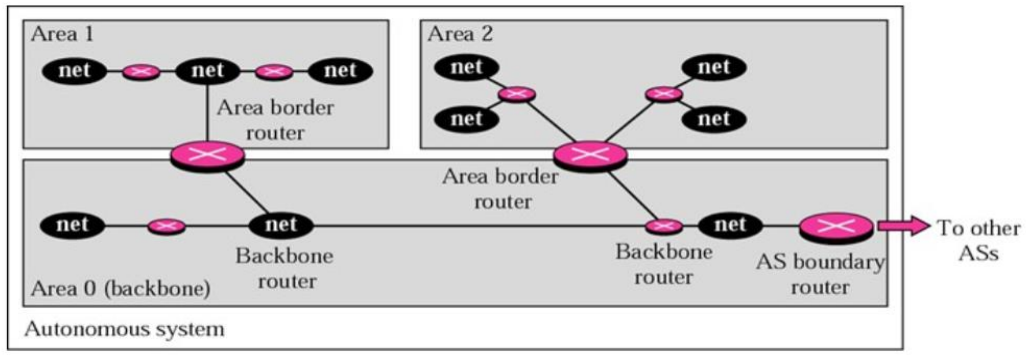
OPEN SHORTEST PATH FIRST(OSPF)

Each link is assigned a weight based on the throughput, round trip time, reliability and so on.

Each OSPF router creates a forwarding table after finding the shortest path tree between itself and the destination using Dijkstra's algorithm.

OSPF is designed to handle routing in a small or large AS.

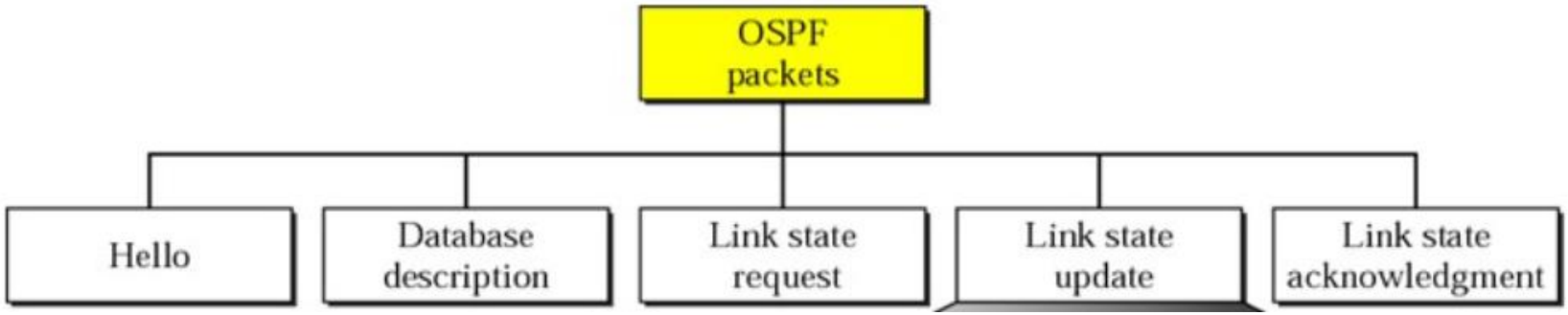
OSPF requires that all routers flood the whole AS with their LSPs to create the global LSDB



OPEN SHORTEST PATH FIRST(OSPF)

Link State Advertisement

OSPF is based on link state routing algorithm, which requires that a router advertise the state of each link to all neighbors for the formation of LSDB.



LINK STATE ROUTING- DIJKSTRAS ALGORITHM

A = 0

B = 4 (A -> B)

C = 5 (A -> C)

D = 4 + 9 = 13 (A -> B -> D)

E = 5 + 3 = 8 (A -> C -> E)

F = 5 + 3 + 6 = 14 (A -> C -> E -> F)