Module 5: Network Layer (Part-4)

Introduction to Networks v7.0 (ITN)

5.13: Routing Algorithms

Distance Vector Routing

1. Distance Vector Routing (DVR)

In this method, each router maintains a table that contains information about the distance and direction to every other node in the network depend on hop count. This table is then shared with other routers in the network. The disadvantage of this method is that it may lead to routing.

What are the metrics used in routing algorithm?

The metrics ensures to determine the best for data packets to reach to destination. There are some metric used in routing algorithm these are:

1. hop count
2. Bandwidth
3. Latency
4. Reliability
5. cost.

Distance Vector Routing

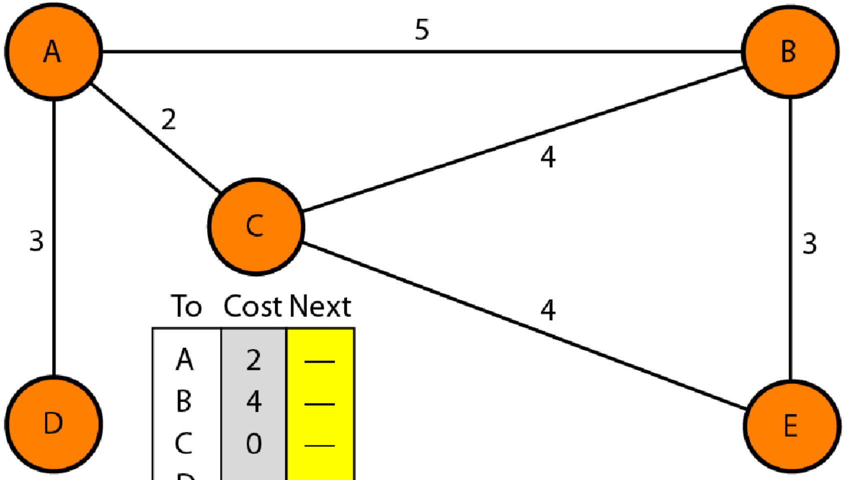
1- *Initialization of tables in distance vector routing (DVR)*

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	∞	
E	3	—

B's table



To	Cost	Next
A	3	—
B	∞	
C	∞	
D	0	—
E	∞	

D's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	∞	
E	4	—

C's table

To	Cost	Next
A	∞	
B	3	—
C	4	—
D	∞	
E	0	—

E's table

Distance Vector Routing

2- *Updating in distance vector routing example: C to A*

From C

To	Cost
A	2
B	4
C	0
D	∞
E	4

Received from C

From A

To	Cost	Next
A	4	C
B	6	C
C	2	C
D	∞	C
E	6	C

A's modified table

A to A via C: ACA = AC+ CA = 2+2

A to B via C: ACB = AC + CB = 2+4

A to C via C: ACC = AC + CC = 2+0

A to D via C: ACD = AC + CD = 2+ inf.

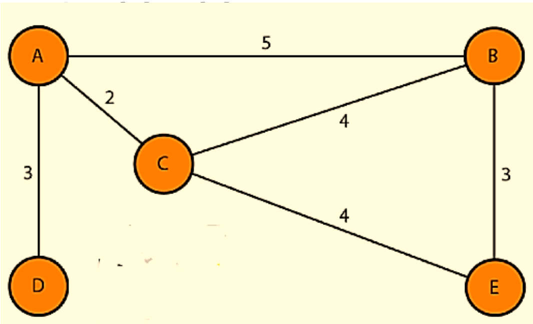
A to E via C: ACD = AC + CE = 2+4

Compare

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's new table



Bellman-Ford Algorithm

Imagine you have a map with different cities connected by roads, each road having a certain distance. The **Bellman–Ford algorithm** is like a guide that helps you find the shortest path from one city to all other cities, even if some roads have negative lengths. It's like a **GPS** for computers, useful for figuring out the quickest way to get from one point to another in a network. The Bell-Ford equation is:

$$\boldsymbol{d_x}(\mathbf{y}) = \mathbf{min}\{\mathbf{c}(\mathbf{x},\mathbf{v}) + \boldsymbol{d_v}(\mathbf{y})\}$$

Where:

$\mathrm{d_x(y)}$: cost of least-cost path from x to y

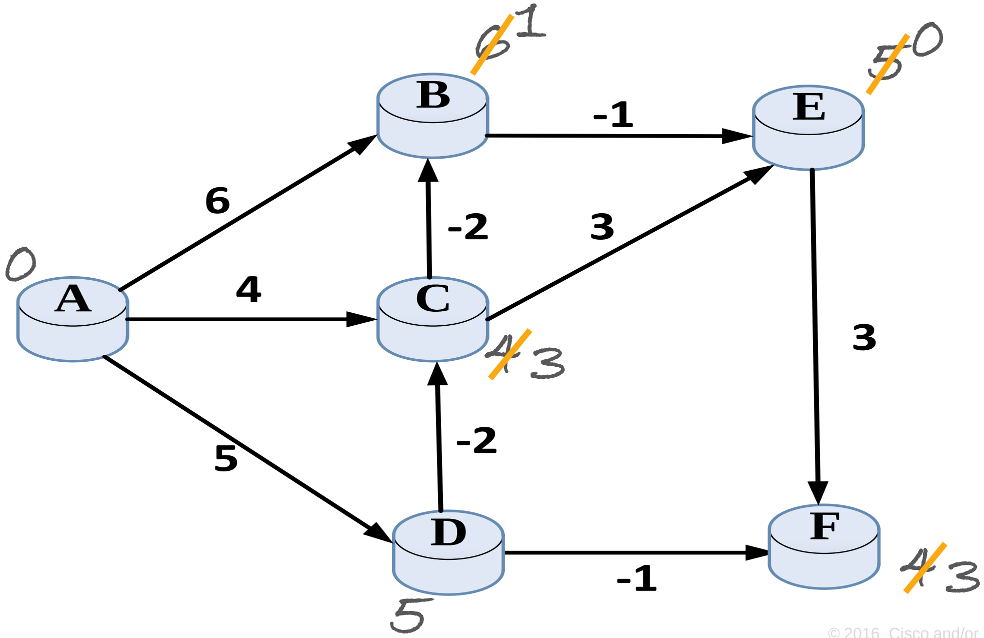
$\mathrm{c(x,v)}$: cost of neighbor v

$\mathrm{d_v(y)}$: cost from neighbor v to destination y

Distance Vector Routing

Example (5-5)

Consider the network shown below. Use Bellman-Ford algorithm to compute the shortest path from node **A** to all other nodes.



Distance Vector Routing

Solution

branches
AB
AC
AD
BE
CE
DC
CB
EF
DF

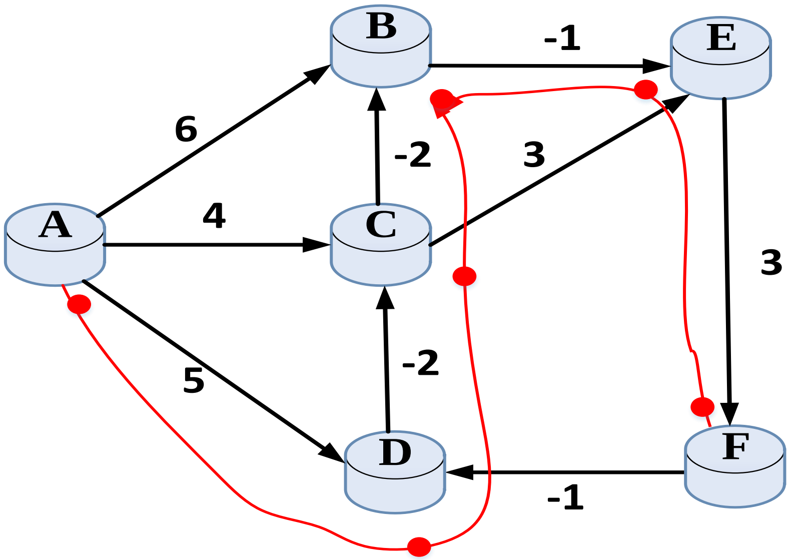
iteration	A	B	C	D	E	F
0	0	∞	∞	∞	∞	∞
1	0	6	4	5	∞	∞
2	0	6	4	5	5	4
3	0	1	3	5	0	3
4	0	1	3	5	0	3
5	0	1	3	5	0	3

الخوارزمية:

1. نعمل جدول يحتوي على عدد النقاط (N) كأعمدة وعدد ال iteration=N-1 كصفوف.
2. نثبت الاوزان على النقط الجارة لنقطة البداية ووضع اوزان لبقية النقاط مساوية ل ∞
3. نثبت بقية الاوزان على بقية النقاط (جيران الجيران) مع ملاحظة اختيار الوزن الاقل في حالة تم التخيير بين اكثر من وزن.
4. نجري العملية (جمع الاوزان وطرحها) على كل النقاط بناء على الفروع المثبته دون تكرار المرور بنفس الفرع بالتكرار الواحد.
5. نلاحظ بعد عدة تكرارات ان الاوزان تثبت على قيمها وبذك ممكن ان نقوم بايقاف العملية واستحصال النتائج

Distance Vector Routing

Solution



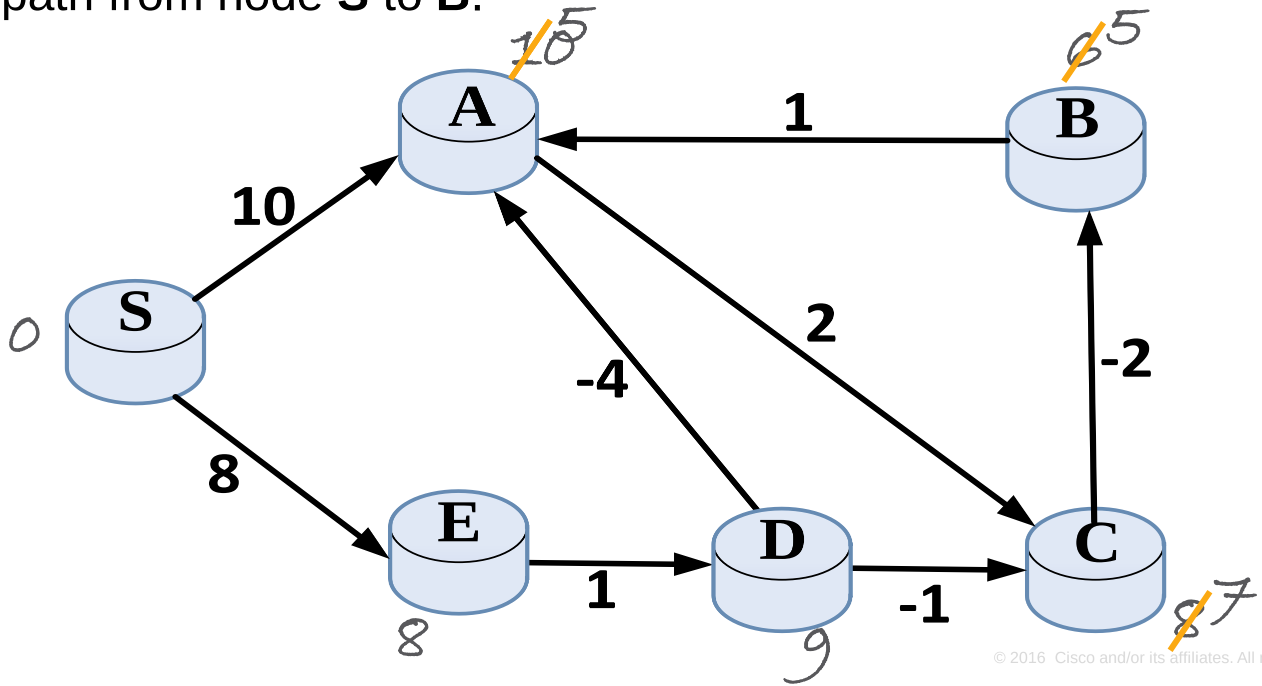
المسار الافضل:

نلاحظ من ال iterationالثالث في الجدول ان القيمة ثبتت لكل نقطة كالاتي:

من Aالى D=5، من Aالى C=3، من Aالى B=1، من Aالى E=0، من Aالى F=3وكما موضح بالشكل اعلاه

Example (5-6)

Consider the network shown below. Use Bellman-Ford algorithm to compute the shortest path from node **S** to **B**.



Distance Vector Routing

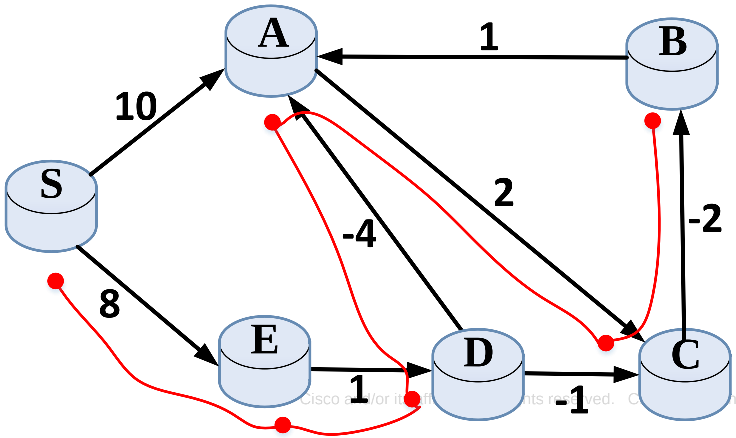
Solution

branches
SA
SE
ED
DA
DC
AC
CB
BA

iteration	S	A	E	D	C	B
0	0	∞	∞	∞	∞	∞
1	0	10	8	∞	∞	∞
2	0	10	8	9	∞	∞
3	0	10	8	9	8	6
4	0	5	8	9	7	5
5	0	5	8	9	7	5

The best path from S to B is:

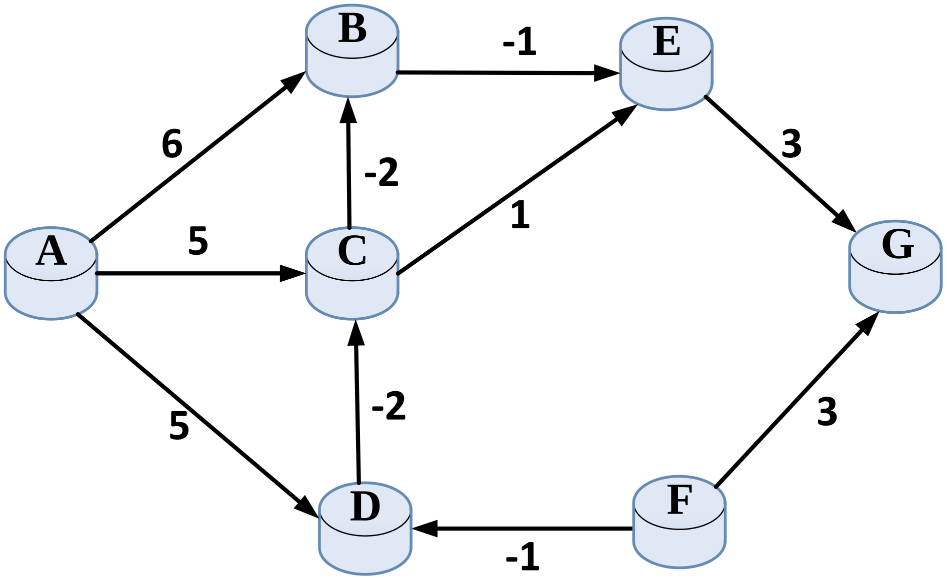
S→E→D→A→C→B



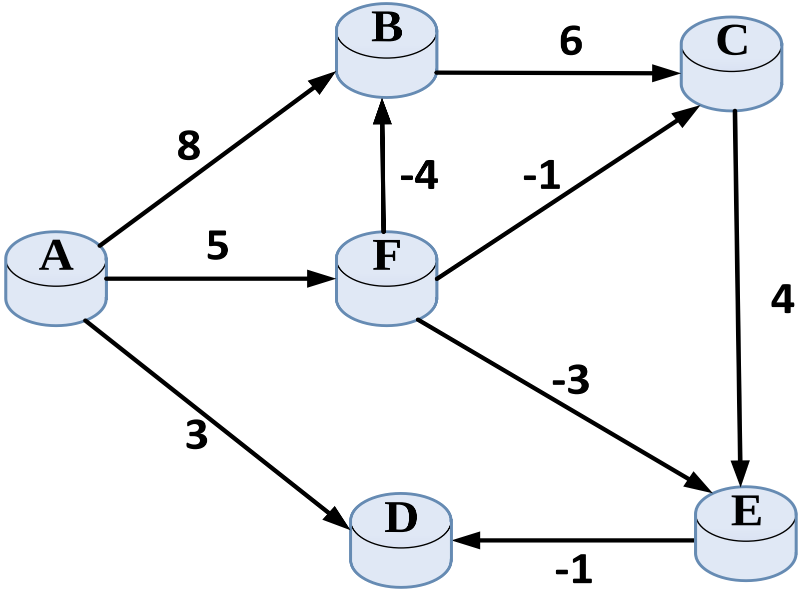
Distance Vector Routing

Homework

Consider the networks shown below. Show the operation of Bellman-Ford to compute the shortest path from node A to all other nodes



Network-1



Network-2

Link state Routing

2- Link State Routing (LSR)

The **primary problem in distance vector** that the algorithm often took **too long to converge** after the network topology changed (due to the count-to-infinity **problem**). Consequently, it was **replaced by** a new algorithm, now called **link state** routing.

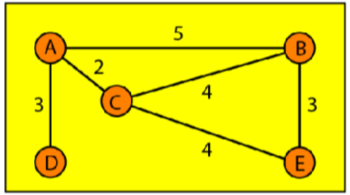
The idea behind link state routing is simple and can be stated as five parts. Each router must:

1. **Discover** its neighbors and learn their network addresses.
2. **Measure** the delay or cost to each of its neighbors.
3. **Construct** a packet telling to all it has just learned.
4. **Send** the packet to all other routers.
5. **Compute** the shortest path to every other router.

Note that the Link-State router tells **ALL** other routers about **ONLY** its neighbors and links.

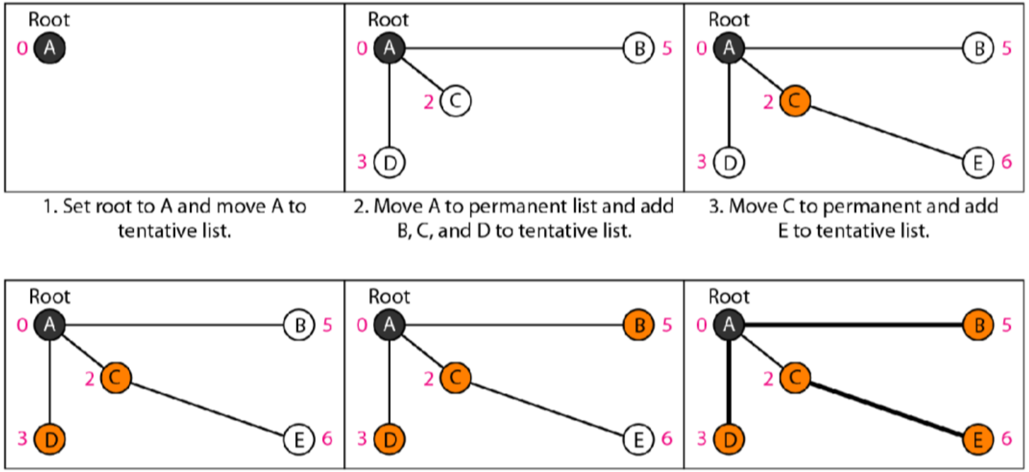
Link State Routing

Example of formation of shortest path tree



Topology

Node	Cost	Next Router
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C



1. Set root to A and move A to tentative list.
2. Move A to permanent list and add B, C, and D to tentative list.
3. Move C to permanent and add E to tentative list.
4. Move D to permanent list.
5. Move B to permanent list.
6. Move E to permanent list (tentative list is empty).

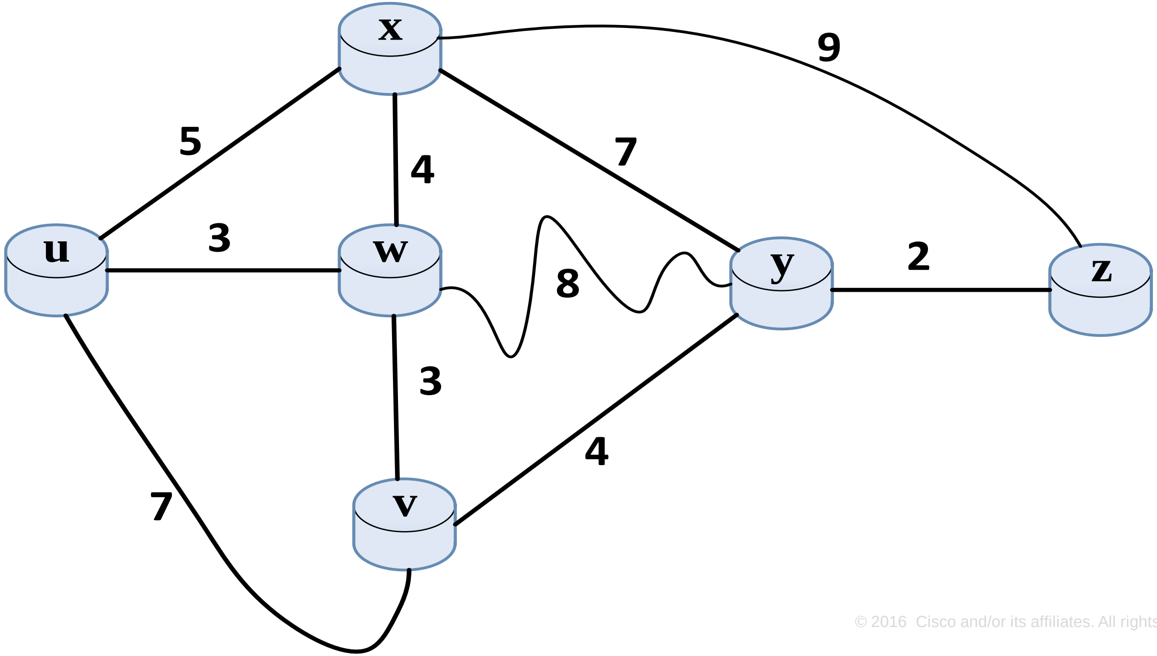
Dijkstra's Algorithm

- **Dijkstra's Algorithm** is a popular type of LSR algorithm that depends on the cost between routers.
  - ➢ In the first row write the whole used nodes in the interior AS.
  - ➢ Compute least cost paths from one node to all other nodes an give forwarding table for that node according to the following rules:
    1. Set the cost, the predecessor node
    2. Use ∞ If there is no direct neighbors
    3. Use --- in the current used nodes
  - ➢ Select the minimum cost and put its node in the next iteration.
  - ➢ Repeat the operation until the whole nodes will documented.
  - ➢ After k iterations, know least cost path to the referred destination.

Link State Routing

Example (5-7)

Consider the network shown below. Show the operation of Dijkstra's algorithm for the computing the shortest path from node **u** through **z**.



Link State Routing

Solution

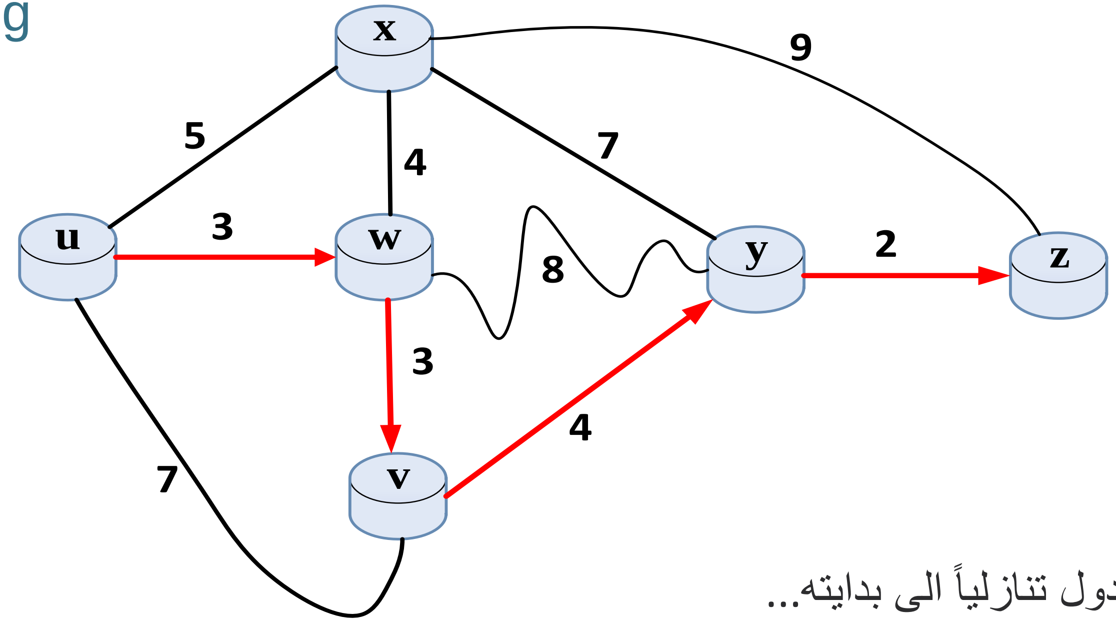
path	u	v	w	x	y	z
[u]	---	7,u	3,u	5,u	∞	∞
3 [w]	---	6,w	---	5,u	11,w	∞
5 [x]	---	6,w	---	---	11,w	14,x
6 [v]	---	---	---	---	10,v	14,x
10 [y]	---	---	---	---	---	12,y
12 [z]	---	---	---	---	---	---

الخوارزمية:

1. نأخذ اول نقطة ثم ناخذ جيرانها مع الاوزان
2. نختار النقطة التي حصلت على اقل وزن ونضعها في بداية السطر التالي مع الوزن
3. نجمع الوزن السابق مع الاوزان التالية ابتداءً من النقطة التي تم اختيارها.. اذا وجدت قيمة اصغر منها فيتم وضعها والا فنبقي القيمة السابقة كما هي
4. النقطة البعيدة عن الجيران تحافظ على قيمتها السابقة لانها افضل من المالانهاية
5. نكرر العملية الى ان نمر بجميع النقاط دون تكرار

Link State Routing

Solution



المسار الافضل:

نبدأ من نهاية الجدول تنازلياً الى بدايته...

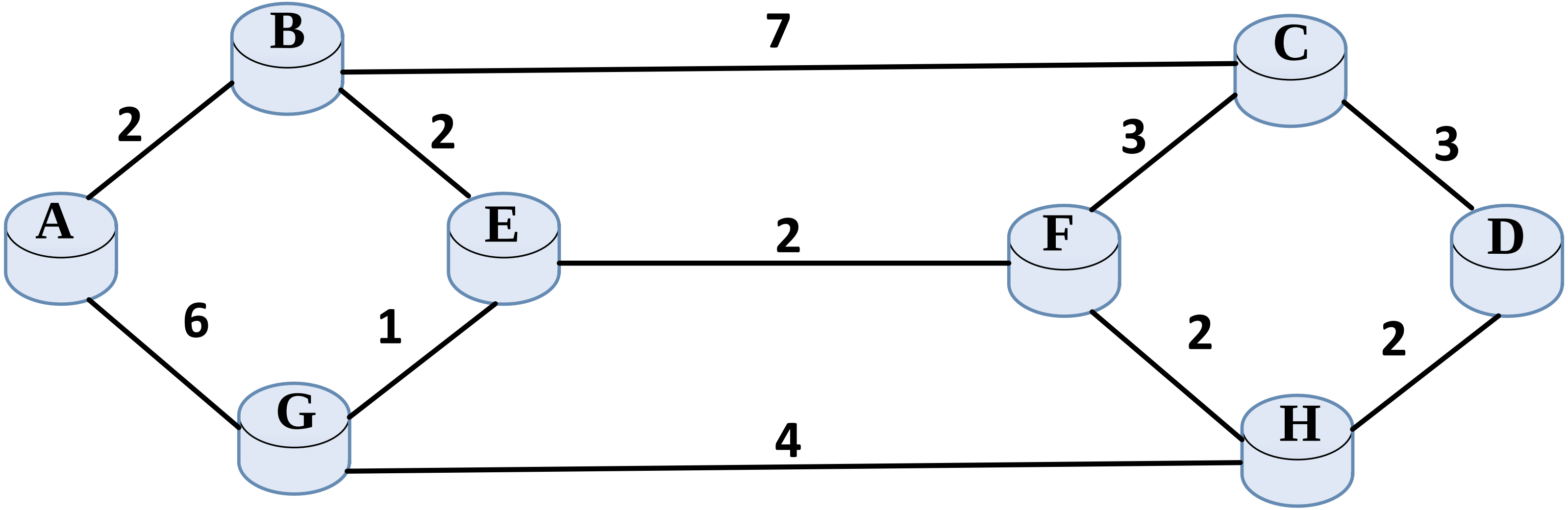
نلاحظ ان اخر قيمة (12,y) تشير الى y، وبعد الانتقال الى y نلاحظ ان اخر قيمة (10,v)تشير الى v، وبعد الانتقال الى v نلاحظ ان اخر قيمة (6,w) تشير الى w، وبعد الانتقال الى w نلاحظ ان اخرقيمة (3,u) تشير الى u ولذلك سيكون افضل مسار هو:

u → w → v → y → z

Link State Routing

Example (5-8)

Consider the network shown below. Show the operation of Dijkstra's algorithm for the computing the shortest path from node A through D.



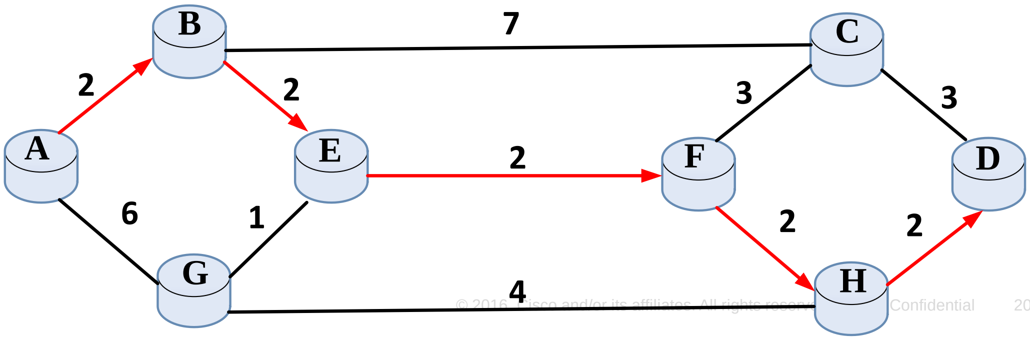
Link State Routing

Solution

	A	B	C	D	E	F	G	H
[A]	---	2,A	∞	∞	∞	∞	6,A	∞
2 [B]	---	---	9,B	∞	4,B	∞	6,A	∞
4 [E]	---	---	9,B	∞	---	6,E	5,E	∞
5 [G]	---	---	9,B	∞	---	6,E	---	9,G
6 [F]	---	---	9,B	∞	---	---	---	8,F
8 [H]	---	---	9,B	10,H	---	---	---	---
10 [C]	---	---	---	10,H	---	---	---	---
10 [D]	---	---	---	---	---	---	---	---

The best path from A to D is:

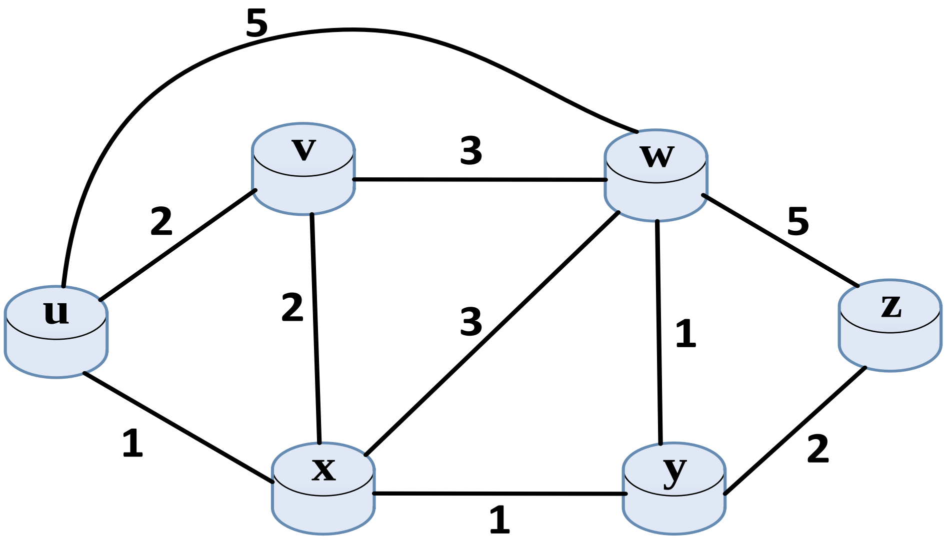
A→B→E→F→H→D



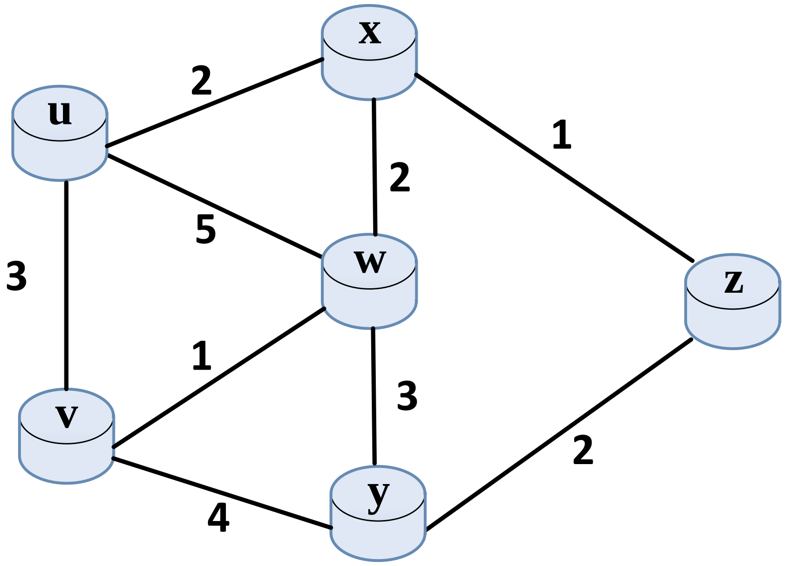
Link State Routing

Homework

Consider the networks shown below. Show the operation of Dijkstra's algorithm for the computing the shortest path from node u to z.



Network-1



Network-2

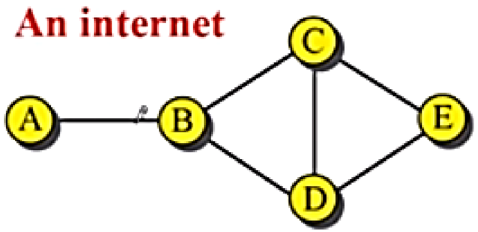
Distance Vector	Link State
• Entire routing table is sent as an update	• Updates are incremental & entire routing table is not sent as update.
• Distance vector protocol send periodic update at every 30 or 90 second	• Updates are triggered not periodic.
• Updates are sent to directly connected neighbor only	• Update are sent to entire network & to just directly connected neighbor.
• Routers don't have end to end visibility of entire network.	• Routers have visibility of entire network of that area only.
• Suffer from count to infinity problem	• No routing loops
. Slow convergence after network topology changed due to the count to infinty problem	• Convergence is fast because of triggered updates.
• Examples: RIP, BGP	• Examples: OSPF, IS-IS
. Easy to configure	. More difficult to configure

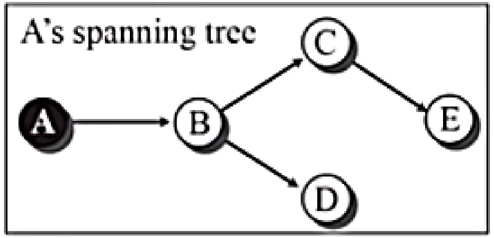
Path Vector Routing

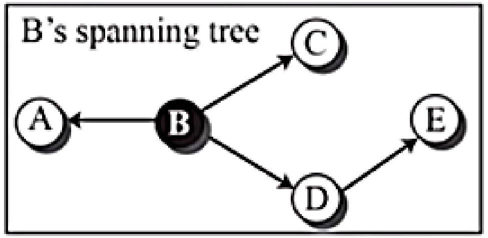
3- Path Vector Routing (PVR) Algorithm

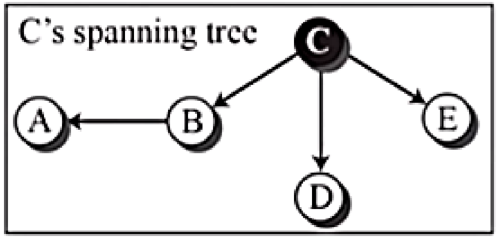
- There is at least one node, called the speaker node, in each AS that creates a routing table and advertises it to speaker nodes in the neighboring AS.
- Only speaker nodes in each AS can communicate with each other. Speaker node advertises the path, not the metric.
- **Sharing:**
  A speaker in an autonomous system shares its table with immediate neighbors.
- **Updating:**
  When a speaker node receives a table from its neighbor, it updates its own table by adding the nodes that are not in its routing table. It also adds its own AS and the AS that sent the table.

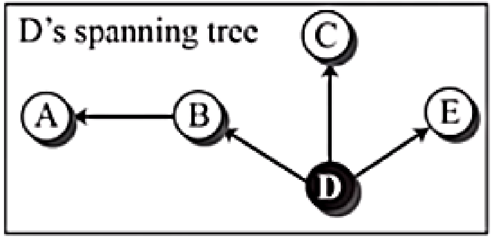
Path Vector Routing

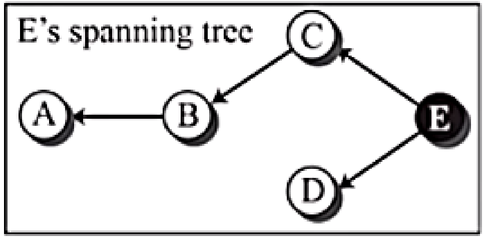










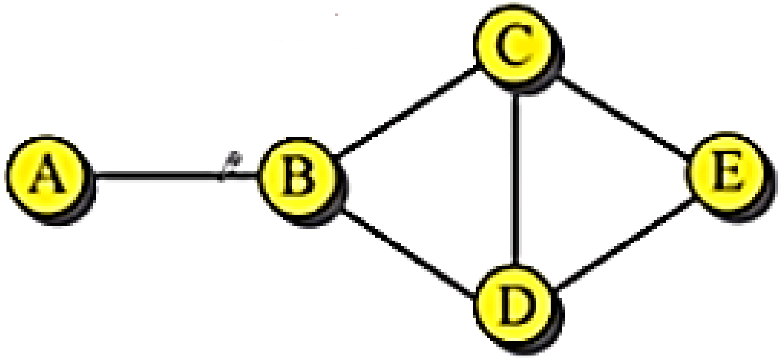


Path Vector Routing

Example (5-9)

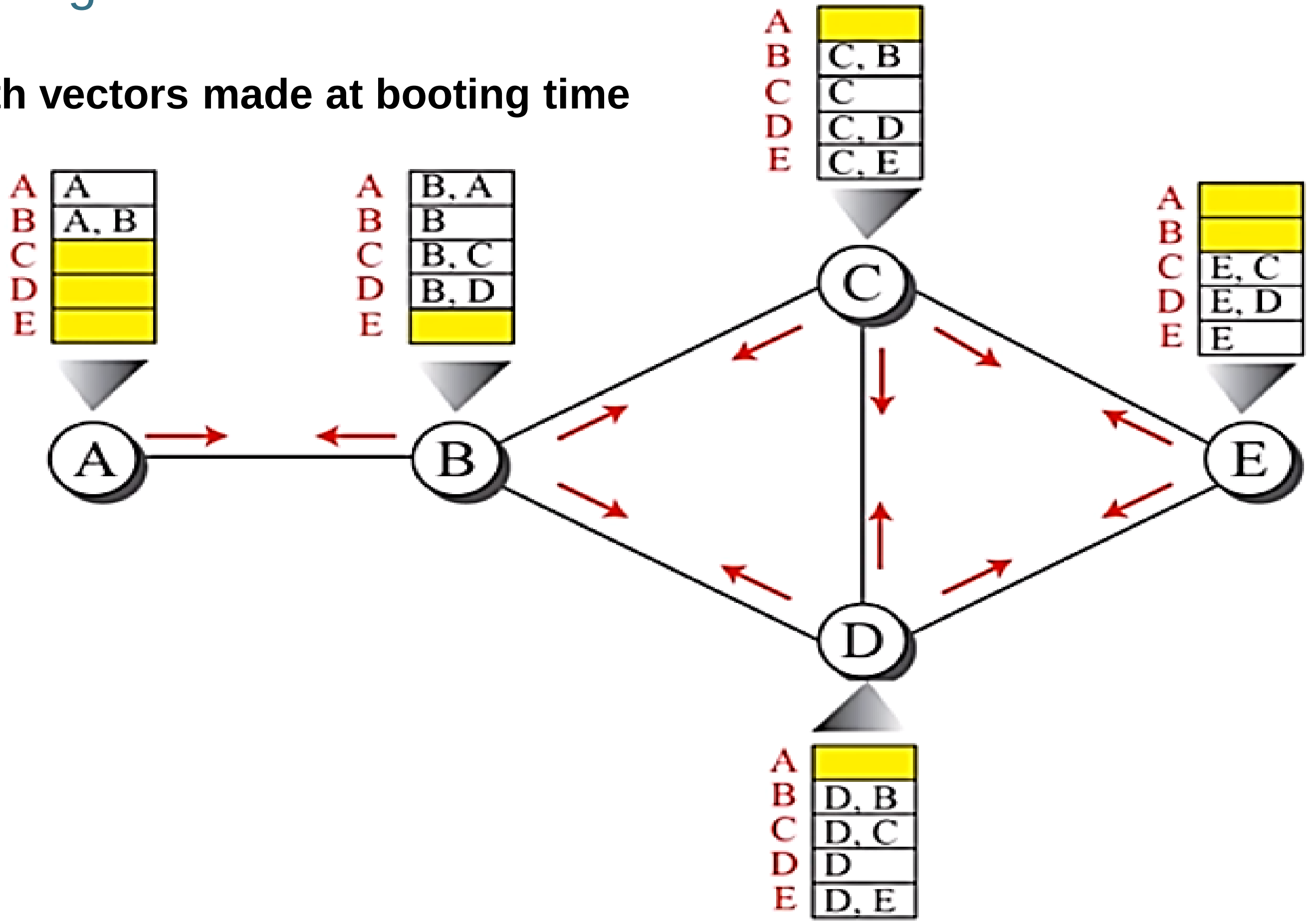
Consider the network shown below to:

1. Made a path vector at booting time for each node.
2. If the speaker node was C, update the path vector for this node.
3. If the speaker node was D, update the path vector for this node (H.W).
4. If the speaker node was E, update the path vector for this node (H.W).



Path Vector Routing

Solution: 1- Path vectors made at booting time



Solution: 2- Updating path vector for node C

B	
A	B, A
B	B
C	B, C
D	B, D
E	

Old C	
A	
B	C, B
C	C
D	C, D
E	C, E

New C	
A	C, B, A
B	C, B
C	C
D	C, D
E	C, E

New C=best (old C , old C+B)

Event 1: C receives a copy of B's vector

D	
A	
B	D, B
C	D, C
D	D
E	D, E

Old C	
A	C, B, A
B	C, B
C	C
D	C, D
E	C, E

New C	
A	C, B, A
B	C, B
C	C
D	C, D
E	C, E

New C=best (old C , old C+D)

Event 2: C receives a copy of D's vector

E	
A	
B	
C	E, C
D	E, D
E	E

Old C	
A	C, B, A
B	C, B
C	C
D	C, D
E	C, E

New C	
A	C, B, A
B	C, B
C	C
D	C, D
E	C, E

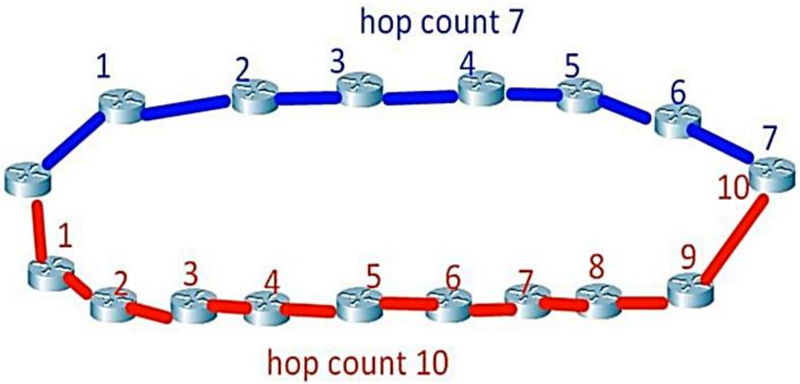
New C=best (old C , old C+E)

Event 3: C receives a copy of E's vector

5.14: Routing Protocols

1- Routing Information Protocol (RIP)

- The router will know about network from the directly connected routers that will tell him the best way (route) to go there.
- There are three types of RIP: RIPv1 and RIPv2 for IPv4, RIPng for IPv6.
- RIP is based on Distance Vector Routing (DVR) algorithm that depends on number of hops between the destination nodes



$$\boxed{Metric = No.of\ Hops}$$

2- Open Shortest Path First (OSPF)

- Every router has the same map so the routers has a full network map
- OSPF divides an AS into different areas.
- OSPF uses **Link State Routing (LSR)** algorithm to update the routing table in each area
- Unlike RIP, OSPF treats the entire network with different philosophy; depends on: Link type, Link cost (metric) and Link condition to define the state of the link.
- Paths between routers depends on the cost which depends on the following relation:

$$\boxed{Metric = \frac{10^8}{Band\ Width\ in\ bps}}$$

Try

Link Speed: 1 Gbps

3- Border Gateway Protocol (BGP)

- BGPv4 is an Exterior Gateway Protocol (EGP) that can exchange routing updates between different Autonomous Systems, so it operate mainly at the border of an AS.
- BGP is not designed to choose paths based on bandwidth, delay and other metrics, but paths are chosen based on policy attributes.
- AS is a collection of networks under a single technical administration, AS is identified by a unique number between 1 – 65535.The range 64512 - 65535 is reserved for private use.
- BGP is a **Path Vector Routing (PVR)** algorithm dependent (advanced distance vector).

4- Enhanced Interior Gateway Routing Protocol (EIGRP)

- EIGRP is a network protocol that enables routers to exchange information more efficiently than earlier network protocols,
- EIGRP is a protocol for routing IPv4 and IPv6.
- EIGRP, however, is a hybrid routing protocol—**it is a distance vector protocol with additional link-state protocol features.**
- Uses triggered updates (EIGRP has **no periodic** updates).
- Provide **fast convergence to minimize network traffic.**
- Uses the **minimum bandwidth** on the path of the destination network, and calculate a route from the total delay metrics.

Routing Protocols
EIGRP: Metric Calculations

$$\text{Metric} = 256^* \left[\left(K_1 * \frac{10^7}{\text{Min. Bandwidth}} + \frac{\dfrac{10^7}{K_2 * \text{Min. Bandwidth}}}{256 - \text{Load}} + \frac{K_3 * \text{Total Delay}}{10}\right) * \frac{K_5}{K_4 + \text{Reliability}}\right]$$

By default, K_1 and K_3 have a value of 1, and K_2, K_4, and K_5 are set to 0. places default K values into the formula and shows a streamlined version of the formula:

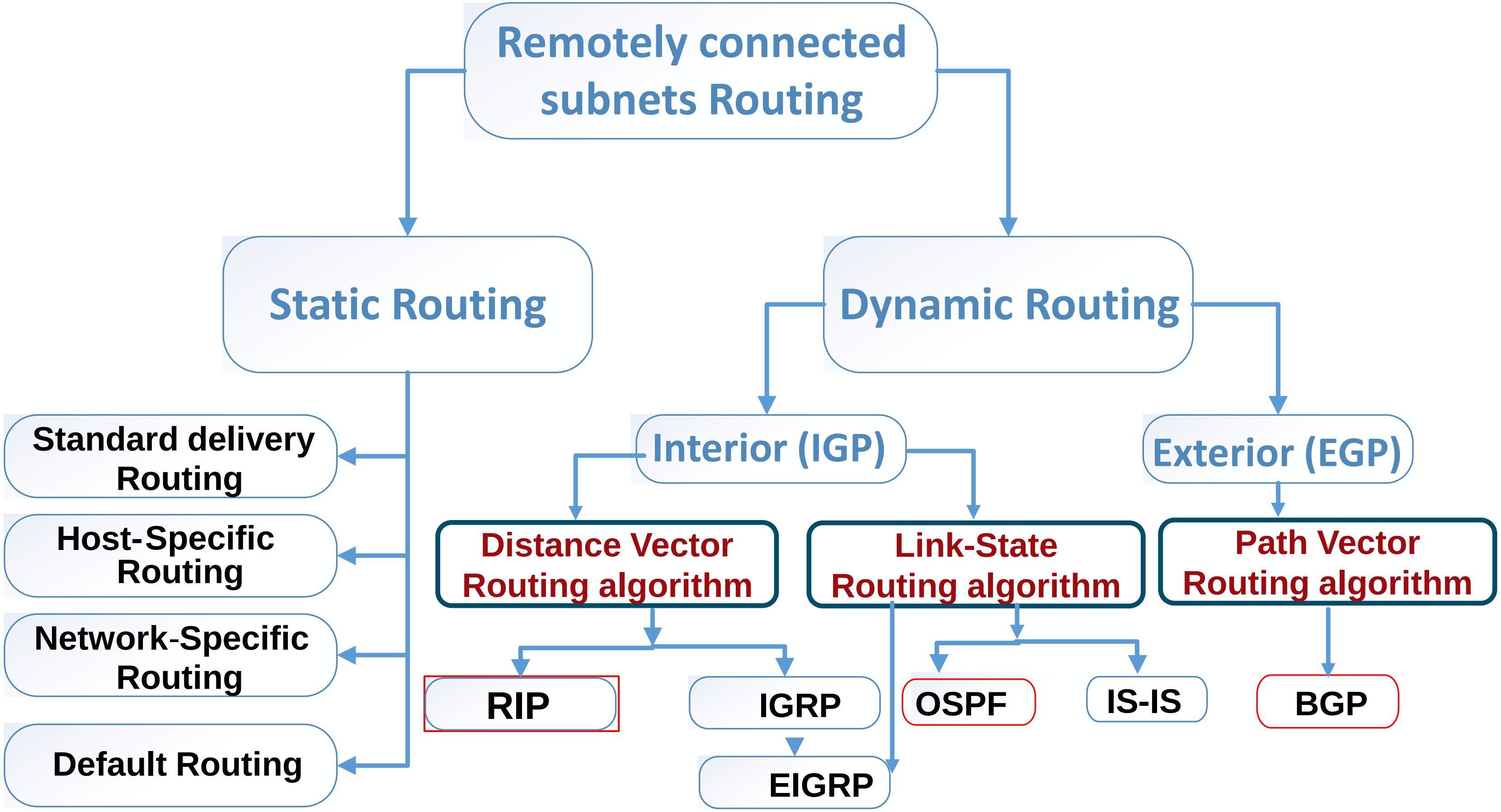
$$\text{Metric} = 256 * \left[\left(1 * \frac{10^7}{\text{Min. Bandwidth}} + \frac{\dfrac{10^7}{0 * \text{Min. Bandwidth}}}{256 - \text{Load}} + \frac{1 * \text{Total Delay}}{10}\right) * \frac{0}{0 + \text{Reliability}}\right]$$

$$\boxed{\text{Metric} = 256 * \left(\frac{10^7}{\text{Min. Bandwidth}} + \frac{\text{Total Delay}}{10}\right)}$$

Try

Link Speed: 1 Gbps
Delay per hop: 10 µs

PROPERTIES	RIP	EIGRP	OSPF
Type	Distance vector	hybrid	Link State
Max hop count	15	100 (by default)	None
Network size	small	Small to large	Small to large
Algorithm	Bellman-ford	Dual	Dijkstra's Algorithm
Convergence	slow	Fast	Fast
VLSM	Yes in v2 not in v1	Yes	Yes
Routing updates	Full table every 30sec	Every 5sec	Only when changes Occur
Bandwidth usage	more	Less	More
Metric	Hop count	Bandwidth , delay	Cost


Remotely connected subnets Routing
Static Routing
Dynamic Routing
Standard delivery Routing
Host-Specific Routing
Network-Specific Routing
Default Routing
Interior (IGP)
Exterior (EGP)
Distance Vector Routing algorithm
Link-State Routing algorithm
Path Vector Routing algorithm
RIP
IGRP
EIGRP
OSPF
IS-IS
BGP