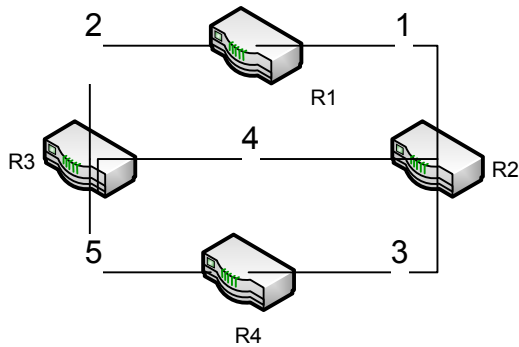


Distance Vector Routing (Bellman – Ford)

Δίνεται ένα παράδειγμα για το πώς δουλεύει ο παραπάνω αλγόριθμος:
Έστω η παρακάτω συνδεσμολογία δικτύου,



με τα αντίστοιχα κόστη να εμφανίζονται πάνω στις γραμμές, τότε ο αλγόριθμος για έναν Router ,έστω τον R1, είναι:

- 1) Οι γειτονικοί Router R2, R3 στέλνουν στον R1 τους πίνακες με τις πληροφορίες που έχουν για κόστη σχετικά με τους γείτονες τους,
- 2) Ο R1 αθροίζει το κόστος που έχει αυτός από τους γείτονες του και παράγει έναν πίνακα με συνολικά κόστη και
- 3) Από τον τελευταίο αυτό πίνακα διαλέγει τα μικρότερα κόστη.

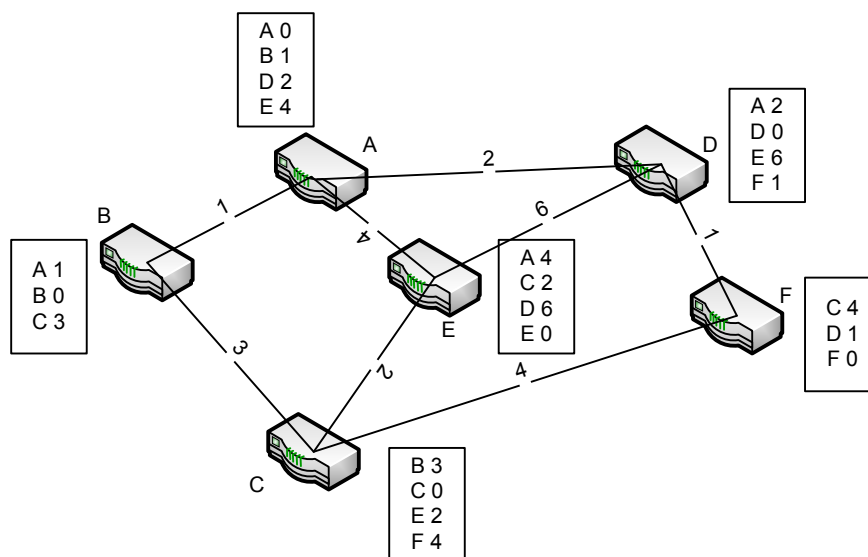
Τα τρία αυτά βήματα φαίνονται στον παρακάτω πίνακα

		Από:	
		R2	R3
Προς:	R1	1	2
	R2	0	4
	R3	4	0
	R4	3	5
Κόστος από R1		1	2

Συνολικό Κόστος	
R2	R3
2	4
1	6
5	2
4	7

Νέος Πίνακας για R1	
0	
1	R2
2	R3
4	R2

Σε ένα πιο σύνθετο πρόβλημα όπως αυτό που φαίνεται παρακάτω έχουμε:



Για τον Router B:

- 1) Ενημέρωση από τους γείτονες του, άθροισμα του κόστους και τελικό πίνακα κόστους.
- 2) Ενημέρωση των γειτόνων του από τους γείτονες τους και επανάληψη της παραπάνω διαδικασίας
- 3) Ενημέρωση του B ξανά από τους γείτονες του με τους αναβαθμισμένους πλέον πίνακες τους και επανάληψη της παραπάνω διαδικασίας

1.

	Από	
Προς:	A	C
A	0	
B	1	3
C		0
D	2	
E	4	2
F		4
Cost	1	3

Total	
A	C
1	-
2	6
-	3
3	-
5	5
-	7

New	
1 via A	
0	
3 via C	
3 via A	
5 via A,C	
7 via C	

2.

A				
		Από		
Προς:	B	E	D	
A	1	4	2	
B	0	-	-	
C	3	2	-	
D	-	6	0	
E	-	0	6	
F	-	-	1	
Cost	1	4	2	

Total		
B	E	D
2	8	4
1	-	-
4	6	-
-	10	2
-	4	8
-	-	3

New	
0	
1 via B	
4 via B	
2 via D	
4 via E	
3 via D	

C				
		Από		
Προς:	B	E	F	
A	1	4	-	
B	0	-	-	
C	3	2	4	
D	-	6	1	
E	-	0	-	
F	-	-	0	
Cost	3	2	4	

Total		
B	E	F
4	6	-
3	-	-
6	4	8
-	8	5
-	2	-
-	-	4

New	
4 via B	
3 via B	
0	
5 via F	
2 via E	
4 via F	

3.

	Από	
Προς:	A	C
A	0	4
B	1	3
C	4	0
D	2	5
E	4	2
F	3	4
Cost	1	3

Total	
A	C
1	7
2	6
5	3
3	8
5	5
4	7

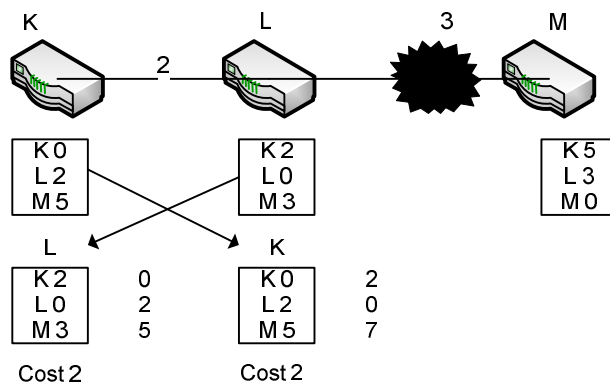
New
1 via A
0
3 via C
3 via A
5 via A,C
4 via A

Αν συγκρίνουμε τώρα τους 2 πίνακες του B δλδ:

New	New
1 via A	1 via A
0	0
3 via C	3 via C
3 via A	3 via A
5 via A,C	5 via A,C
7 via C	4 via A

Παρατηρούμε ότι το κόστος για τον F έχει αλλάξει από 7 σε 4 και είναι μέσω του Router A αντί του Router C.

Ένα πρόβλημα που παρατηρείτε στον παραπάνω αλγόριθμο είναι όταν σπάσει μια σύνδεση και φαίνεται στο παρακάτω σχήμα:



Δηλαδή ενώ έχει σπάσει η σύνδεση ο K βλέπει δρόμο για τον M μέσω του L με κόστος 5 και ο L μέσω του K με κόστος 7. Αυτό θα συνεχιστεί μέχρι το κόστος να γίνει τόσο υψηλό που θα είναι εκτός πραγματικότητας, τότε μόνο θα γίνει αναδιαπραγμάτευση.

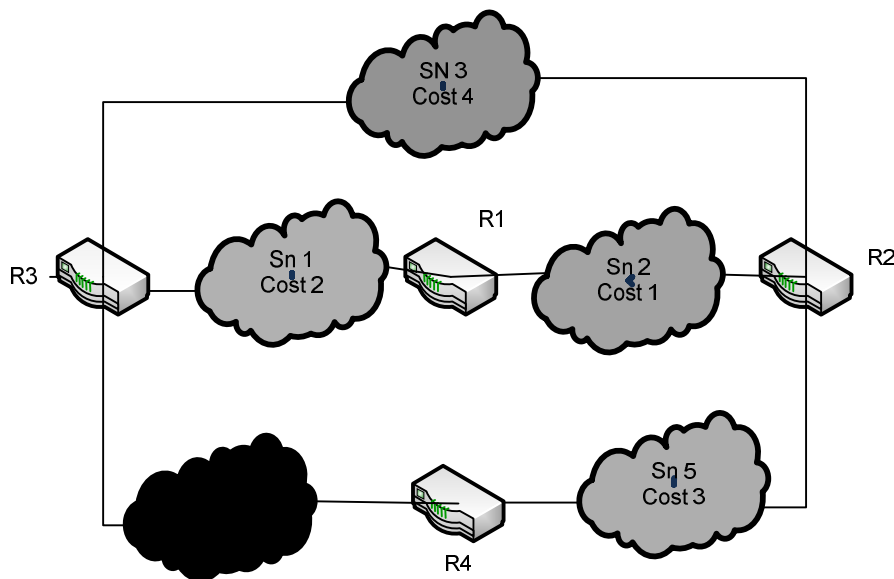
Link State.

Ο παραπάνω αλγόριθμος μας δίνει πληροφορίες για όλες τις σύντομες διαδρομές χωρίς όμως να δίνει πληροφορίες για την τοπολογία του δικτύου. Ο Link state δίνει πληροφορίες για την τοπολογία του δικτύου χωρίς όμως να μπορεί να καθορίσει τις καλύτερες διαδρομές. Η φιλοσοφία είναι ότι οι πληροφορίες που έρχονται από έναν γείτονα αναμεταδίδονται σε όλους τους άλλους που συνδέονται πάνω στον δρομολογητή εκτός από αυτόν τον οποίο ήρθαν οι πληροφορίες αυτές.

Παράδειγμα:

Link state data received from	Link state data		
	Link to Router	Via Network	Cost of network
R1	R2	Netid 2	1
	R3	Netid 1	2
R2	R1	Netid 2	1
	R3	Netid 3	4
	R4	Netid 5	3
R3	R2	Netid 3	4
	R4	Netid 4	5
	R1	Netid 1	2
R4	R2	Netid 5	3
	R3	Netid 4	5

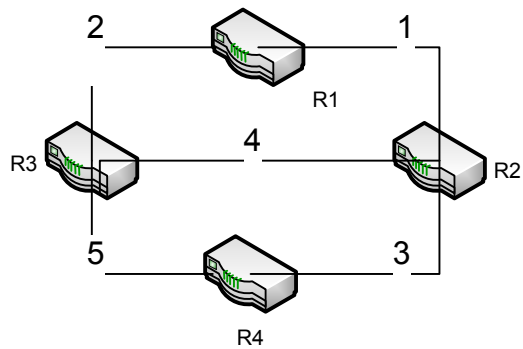
Ο παραπάνω πίνακας δίνει την παρακάτω τοπολογία και συνδεσμολογία:



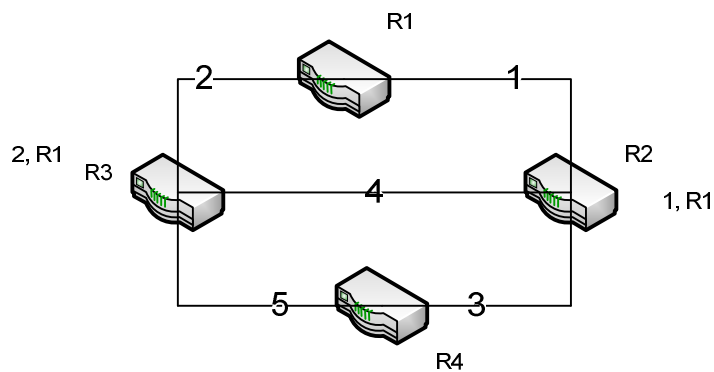
Το πρόβλημα με αυτόν τον αλγόριθμο είναι ότι κάθε φορά που ανταλλάσσονται πληροφορίες μεταξύ των δρομολογητών απαιτούνται μεγάλο ποσοστό πόρων δικτύου, γιατί είναι και ένας αλγόριθμος ο οποίος δεν τρέχει πολύ συχνά.

Τώρα για να βρούμε την καλύτερη δυνατή διαδρομή χρησιμοποιείτε είτε ο Bellman – Ford είτε ο dijkstra, ο οποίος είναι:

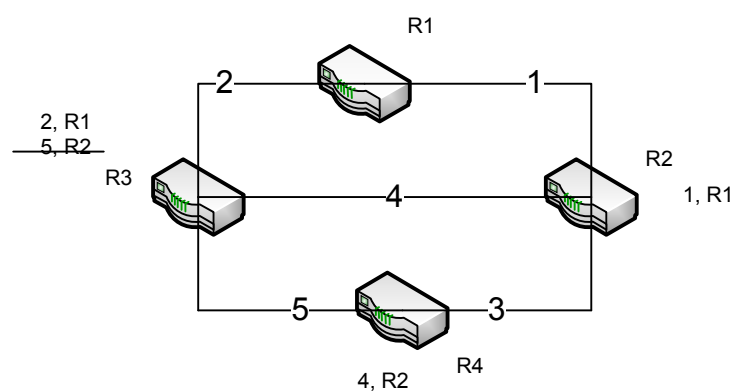
Παίρνουμε το παραπάνω σχήμα χρησιμοποιώντας όμως μόνο τα κόστη των γραμμών και όχι τα ενδιάμεσα δίκτυα.



Θεωρούμε σαν αναφορά τον R1 τότε οι δύο γείτονες του υπολογίζουν το κόστος από αυτούς προς τον R1 και έχουμε το παρακάτω σχήμα:

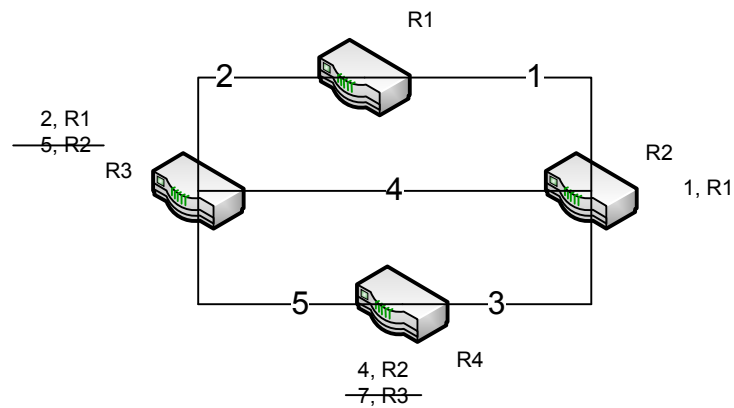


Στη συνέχεια υπολογίζετε το κόστος των υπολοίπων router μέσω εκείνου που έχει το μικρότερο κόστος προς τον R1, δηλ τον R2



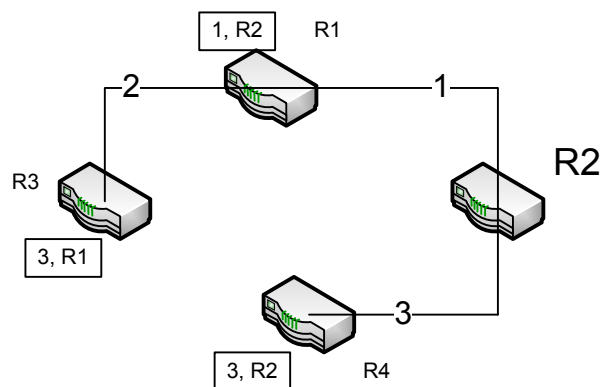
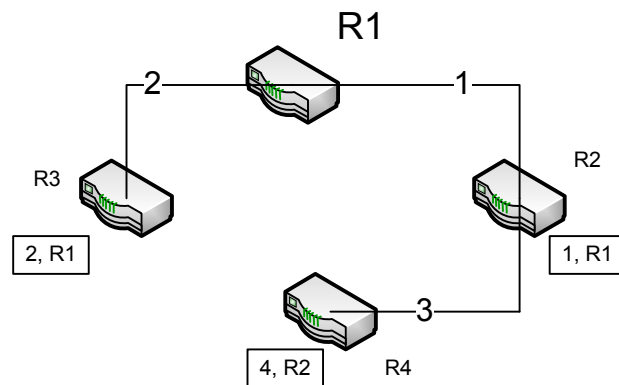
Και ουσιαστικά επιλέγετε το χαμηλότερο κόστος.

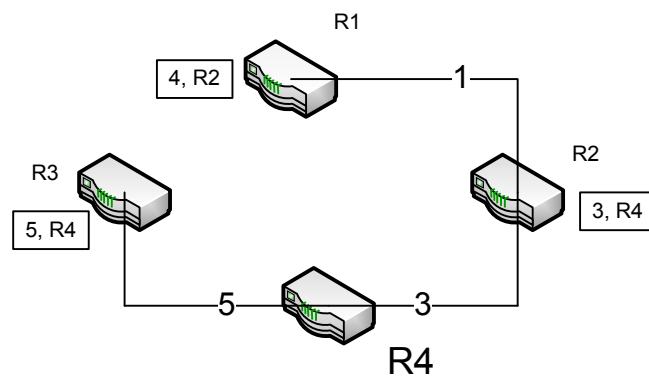
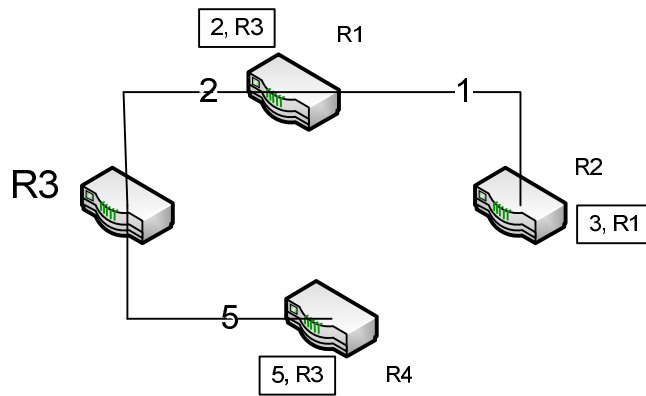
Στη συνέχεια επαναλαμβάνετε η διαδικασία για τον επόμενο γείτονα R3



Υπολογίσαμε τις καλύτερες διαδρομές προς τον R1 με σημείο αναφοράς τον R1.

Τώρα αν πάρουμε σαν σημείο αναφοράς κάθε Router ξεχωριστά τότε προκύπτουν τα παρακάτω σχήματα:





Το ότι δεν φαίνονται οι υπόλοιπες διαδρομές δεν σημαίνει ότι δεν υπάρχουν, απλά δεν χρησιμοποιούνται σε επιλογές καλύτερης δρομολόγησης από τους Router, θα χρησιμοποιηθούν αν υπάρξει κάποιο πρόβλημα και επαναυπολογισθούν οι καλύτερες δυνατές διαδρομές.

Τελικός Πίνακας:

	R1	R2	R3	R4
R1	x	1,R1	2,R1	4,R2
R2	1,R2	x	3,R1	3,R2
R3	2,R3		x	5,R3
R4	4,R2	3,R4	5,R4	x