È quindi necessario evitare che lo stesso pacchetto venga inoltrato all'infinito, a tal fine si introducono due semplici meccanismi di controllo:

A viene stabilito dal mittente un numero massimo di hop che il pacchetto può fare e viene inserito in un contatore che viene decrementato a ogni hop fino a giungere a 0;

B ogni router tiene memoria di ogni pacchetto che instrada, in modo tale che se lo riceve per la seconda volta lo ignora (ogni router per ogni pacchetto tiene una coppia di dati: *source router ID, sequence number*).

Il vantaggio della tecnica di flooding come *algoritmo di instradamento* è l'estrema semplicità dato che l'elaborazione eseguita in ciascun nodo è praticamente nulla: è particolarmente adatto alla trasmissione broadcasting dove l'informazione deve proprio essere inviata a tutti gli host.
Un altro campo nel quale viene utilizzato è quello militare, dove è necessario avere la massima sicurezza, massima affidabilità e robustezza anche a scapito dell'efficienza.

Questo algoritmo risulta essere il più semplice ed efficace in quanto trova sempre il percorso migliore nel minor tempo possibile ma, naturalmente, a scapito della efficienza dato che tende a saturare i canali trasmissivi: per questo motivo viene utilizzato come termine di paragone per gli altri algoritmi di ricerca del cammino ottimo.

Flow-based routing

Questo algoritmo calcola il percorso migliore in base al traffico medio di ogni linea, ipotizzandolo "abbastanza" costante nel tempo. I valori di ritardo per ogni linea vengono calcolati conoscendo le caratteristiche fisiche della linea stessa e la stima del traffico medio per ogni coppia di router.

Questo metodo è simile a quello che ogni automobilista utilizza per raggiungere una destinazione evitando le strade dove sa già, per esperienza, che saranno trafficate: se però tutti utilizzassero questo metodo di fatto nessuno passerebbe sulle strade trafficate, che diventerebbero deserte, e tutti gli automobilisti finirebbero "in coda" su strade alternative.

L'algoritmo di Dijkstra

L'algoritmo di Dijkstra è un algoritmo che rientra tra i shortest path routing e permette di calcolare l'albero dei cammini minimi tra un nodo di un grafo e tutti gli altri in modo da configurare le tabelle di routing: data la sua natura statica se cambia la configurazione della rete è necessario ricalcolare l'albero dei cammini minimi ripartendo da capo.
È un algoritmo di tipo centralizzato in quanto ciascun nodo calcola il cammino ottimo da se stesso verso tutti gli altri nodi in modo indipendente, partendo unicamente dalle informazioni complete sulla topologia della rete e sulle caratteristiche dei link.

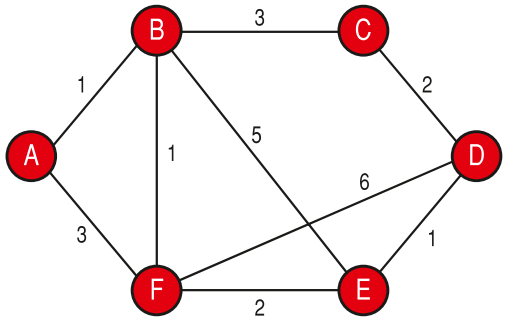
Prima di vedere come procede, definiamo innanzi tutto l'obiettivo che è quello di trovare i cammini minimi tra un nodo 1 (sorgente) e tutti gli altri nodi partendo dall'ipotesi che tutti gli archi abbiamo pesi positivi.

Il metodo di Dijkstra si assegna ai nodi delle etichette, che possono essere:
- temporanee: il costo massimo per la raggiungibilità del nodo in esame;
- permanenti: costo del cammino minimo.

L'algoritmo di calcolo modifica le etichette temporanee cercando di minimizzarle e di renderle permanenti.

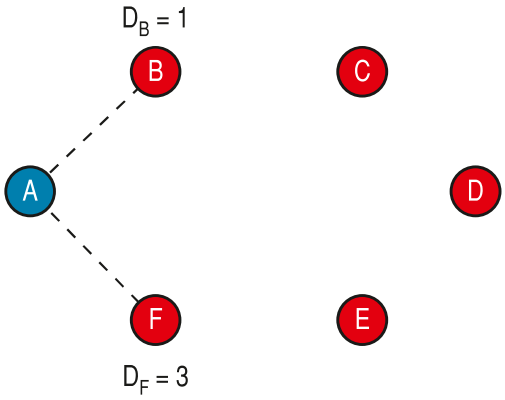
ESEMPIO 8

Vediamo un semplice esempio applicando l'algoritmo al grafo non orientato della figura: ▶



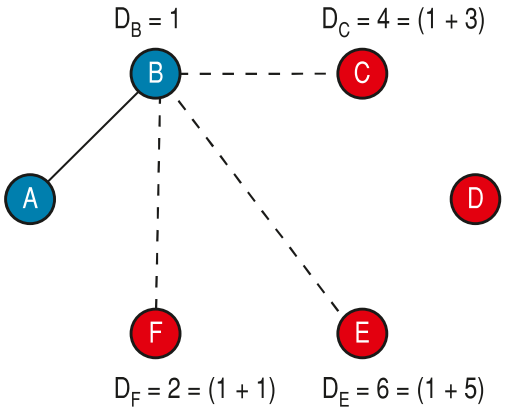
Scegliamo "a piacere" un nodo, in quanto il risultato che otteniamo è ininfluente dal nodo di partenza, e iniziamo a scrivere le etichette provvisorie sui due nodi a esso adiacenti.

Partiamo da A ed etichettiamo B con 1 e F con 3. ▶

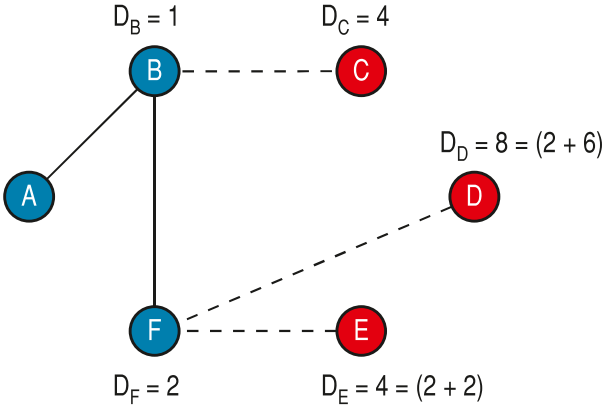


Scegliamo ora il "percorso meno costoso" che si è individuato, cioè quello che parte dal nodo B, e ripetiamo le stesse operazioni assegnando le etichette provvisorie ai nodi C, F ed E ottenute come somma dei percorsi dal nodo precedente (1) e dei nuovi pesi dei rispettivi archi: ▶

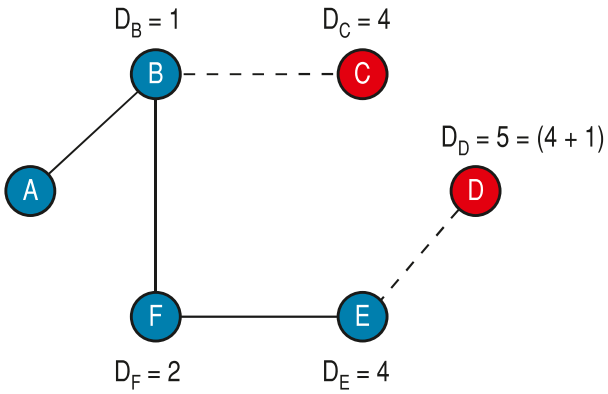
> La scelta secondo la quale il nodo successivo da visitare è quello più vicino a uno dei nodi già visitati dà il nome all'algoritmo Shortest Path First, cioè prima il percorso più breve.



Procediamo in modo analogo scegliendo il percorso più corto tra i 3 sino a ora individuati, cioè quello che ha raggiunto il nodo F (peso 2), e aggiorniamo le etichette dei nodi a esso adiacenti ($D_D = 2 + 6 = 8$, $D_E = 2 + 2$): ▶

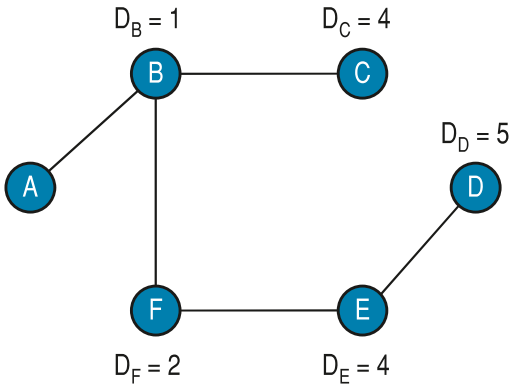


Con lo stesso criterio il prossimo nodo da analizzare è il nodo E (peso 4): dal nodo E possiamo raggiungere il nodo D con peso D_D=4+1=5 che è minore del precedente valore dell'etichetta provvisoria: la sostituiamo con una *nuova etichetta provvisoria*. ▶



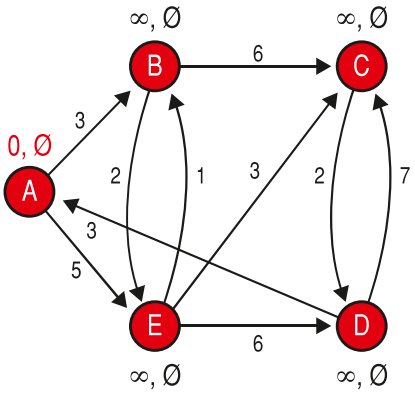
Il nodo con peso minore tra quelli che non sono ancora sati analizzati è il C che ha peso 4, ma non porta migliorie in quanto la sua distanza da D ha peso 2 che peggiorerebbe l'etichetta parziale.

Ultimo nodo è il nodo D, ma anch'esso non introduce migliorie e quindi le etichette provvisorie diventano permanenti e si ottiene il seguente albero minimo: ▶

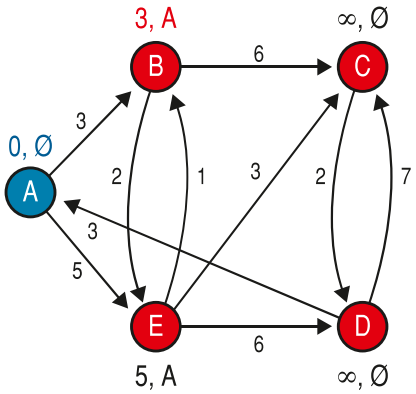


ESEMPIO 9

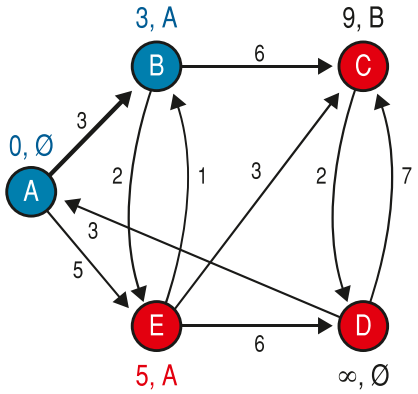
Vediamo ora un secondo esempio in un grafo orientato dove all'etichetta provvisoria che indica il peso aggiungiamo un secondo dato che indica il nodo antecedente che lo ha raggiunto: all'inizio le etichette vengono poste a ∞ e il nodo di provenienza viene indicato con il valore 0. ▶



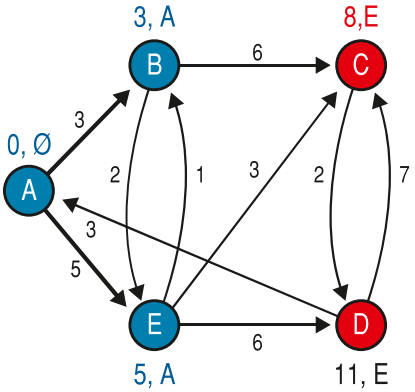
Partendo da A etichettiamo i nodi B(3,A) e E(5,A). ▶



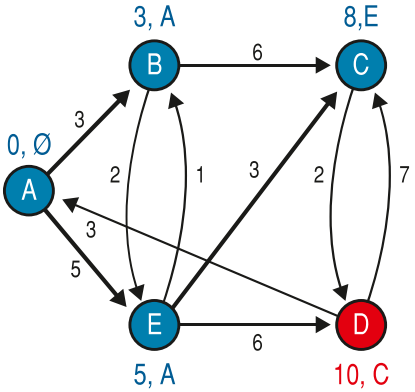
Il nodo successivo è B che è adiacente a C(3+6,B) e a E(3+2, ?), che è già etichettato, e il nuovo peso 5 non migliora quello provvisorio che ha nella sua etichetta, e quindi lo lasciamo invariato. ▶



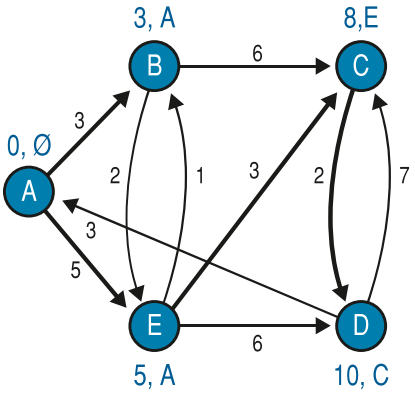
Il nuovo percorso minimo è quello che raggiunge il nodo E, quindi analizziamo i suoi nodi adiacenti C(5+3,E), che migliora la precedente etichetta e quindi viene aggiornata, e D(5+6,E) che è la prima etichetta provvisoria per il nodo D. ▶



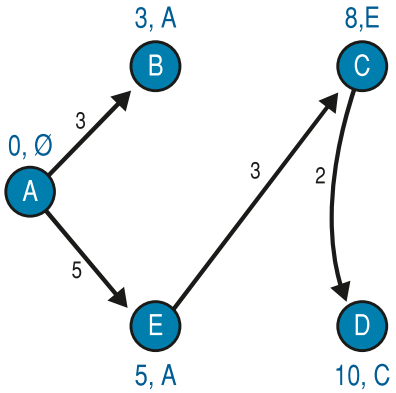
Tra i nodi D e C il Shortest Path è quello che raggiunge il nodo C, e attraverso esso si migliora la raggiungibilità del nodo con D(8+2,C). ▶



Ultimo a essere analizzato è il nodo D che non introduce migliorie, quindi il risultato è il seguente: ▶



e cancellando gli archi "non utilizzati" otteniamo il seguente albero minimo ▶



Zoom su...

PSEUDOCODIFICA

Vediamo un esempio di codifica in linguaggio di progetto dell'algoritmo di Dijkstra, dopo aver introdotto la notazione che verrà utilizzata per comodità nello pseudocodice:

- c(x,y): costo del collegamento dal nodo x al nodo y; è posto a ∞ se non sono adiacenti;
- D(v): costo del cammino dal nodo origine alla destinazione *v* per quanto riguarda l'iterazione corrente;
- p(v): immediato predecessore di *v* lungo il cammino;
- N': sottoinsieme di nodi per cui il cammino a costo minimo dall'origine è definitivamente noto;
- N: insieme di tutti i nodi presenti nel grafo.

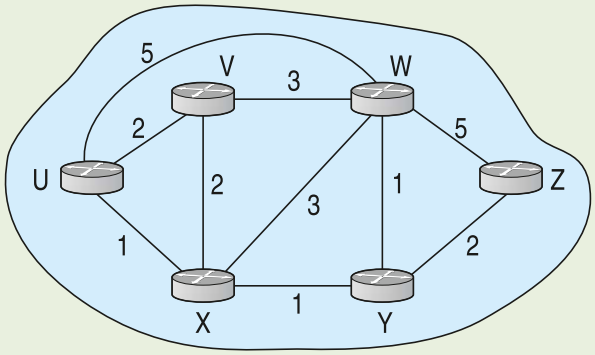
```
 1 Inizializzazione
 2 N' = {u}
 3 per tutti i nodi v // oppure finché N' = N
 4  se v è adiacente a u
 5    allora D(v) = c(u,v)
 6    altrimenti D(v) = ∞
 7   ripeti
 8    determina un w non in N' tale che D(w) sia minimo
 9  aggiungi w a N'
10  per ciascun nodo v adiacente a w e non in N'
11    aggiorna D(v) = min(D(v), D(w) + c(w,v))
```

Dove con l'istruzione

```
11 D(v) = min(D(v), D(w) + c(w,v))
```

si assegna a D(v) il nuovo costo verso v che è il valore minimo tra il vecchio costo presente nella etichetta temporanea oppure il costo del cammino minimo noto verso w più il costo da w a v.

Lo applichiamo alla seguente rete ▶



ottenendo: ▼

Passo	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					

Conclusioni

L'algoritmo di Dijkstra individua il cammino a lunghezza minima tra un nodo e tutti gli altri nodi di un grafo G procedendo per passi successivi e terminando quando si sono esplorati tutti i nodi e si è ottenuto uno spanning tree contenente i cammini a costo minimo tra il nodo sorgente e tutti gli altri nodi del grafo.

È doveroso ricordare che per poter effettuare tali calcoli è necessario avere a disposizione tutte le informazioni sulla topologia della rete.

La complessità di calcolo di questo algoritmo è $O(n^2)$, così ottenuta dal calcolo del numero di etichette da calcolare: $(n-1) + (n-2) + (n-3) + \ldots + 2 + 1 = n(n-2)/2$.
Esistono anche implementazioni che, grazie a ottimizzazioni, hanno complessità $O(n \cdot \log n)$.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **La classificazione degli algoritmi di routing in base alla topologia prevede:**

a) algoritmi distribuiti, isolati, centralizzati
b) algoritmi locali, remoti, isolati
c) algoritmi distribuiti, isolati, locali
d) algoritmi remoti, isolati, centralizzati

2 **Gli algoritmi di routing statici possono inoltre essere classificati per tipo in tre gruppi:**

a) shortest path routing
b) flooding
c) link state
d) flow-based routing

3 **Il routing statico:**

a) prevede che il router sia preconfigurato
b) prevede che l'amministratore configuri il router
c) prevede che la rete non subisca variazioni per l'intera durata della configurazione del router
d) prevede che la rete non subisca variazioni per l'intera vita della rete

4 **In che cosa consiste l'hot potato?**

a) Un protocollo utilizzato dai bridge
b) Un arbitraggio delle reti locali
c) Un algoritmo di routing isolato
d) Un algoritmo di routing dinamico

5 **Un Link State Packet contiene i seguenti dati (indicare quelli non presenti):**

a) stato di ogni link connesso al router
b) identità di ogni vicino connesso all'altro estremo del link
c) costo del link
d) numero di host adicenti
e) numero di sequenza per l'LSP
f) numero di pacchetti nella sequenza
g) checksum
h) lifetime

6 **Tra i criteri di minimizzazione dei percorsi possiamo utilizzare:**

a) la lunghezza fisica dei collegamenti
b) il tempo medio di trasmissione
c) una combinazione di numero di messaggi, banda trasmissiva, dimensione del messaggio
d) il numero di hop

Test vero/falso

1	Gli algoritmi di routing statici sono di tipo adattivo.	V F
2	Tra gli algoritmi di routing statici è incluso il metodo manuale.	V F
3	La configurazione manuale viene comunque sempre utilizzata negli end-host.	V F
4	Con sink tree si intende l'insieme dei cammini ottimi da tutti i router a uno specifico router.	V F
5	La tecnica del flooding consiste nell'inviare ogni pacchetto su tutte le porte di uscita.	V F
6	La tecnica del flooding è particolarmente adatta alla trasmissione broadcasting.	V F
7	L'algoritmo di calcolo modifica le etichette temporanee per minimizzarle e renderle permanenti.	V F
8	Il metodo di Dijkstra si utilizza solo sui grafi non orientati.	V F