Reti di Calcolatori

Prof. Roberto Canonico
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

Corso di Laurea in Ingegneria Informatica

Routing Link-State ed algoritmo di Dijkstra

I lucidi presentati al corso sono uno strumento didattico che NON sostituisce i testi indicati nel programma del corso

Università degli Studi di Napoli Federico II

Nota di copyright per le slide COMICS

Nota di Copyright

Autori:
Simon Pietro Romano, Antonio Pescapè, Stefano Avallone, Marcello Esposito, Roberto Canonico, Giorgio Ventre

Università degli Studi di Napoli Federico II

Algoritmi di routing Link State

- Ogni router:
  - impara il suo ambito locale (linee e nodi adiacenti)
  - trasmette queste informazioni a tutti gli altri router della rete tramite un *Link State Packet* (LSP)
  - memorizza gli LSP trasmessi dagli altri router e costruisce una mappa della rete
  - Calcola, in maniera indipendente, le sue tabelle di instradamento applicando alla mappa della rete l'algoritmo di Dijkstra, noto come *Shortest Path First* (SPF)
- Tale approccio è utilizzato nello standard ISO 10589 (protocollo IS-IS) e nel protocollo OSPF (adottato in reti TCP/IP)

Il processo di *update*

- Ogni router genera un Link State Packet (LSP) contenente:
  - stato di ogni link connesso al router
  - identità di ogni vicino connesso all'altro estremo del link
  - costo del link
  - numero di sequenza per l'LSP
  - checksum
  - Lifetime:
    - la validità di ogni LSP è limitata nel tempo (e.g. un errore sul numero di sequenza potrebbe rendere un LSP valido per anni)

LSP flooding

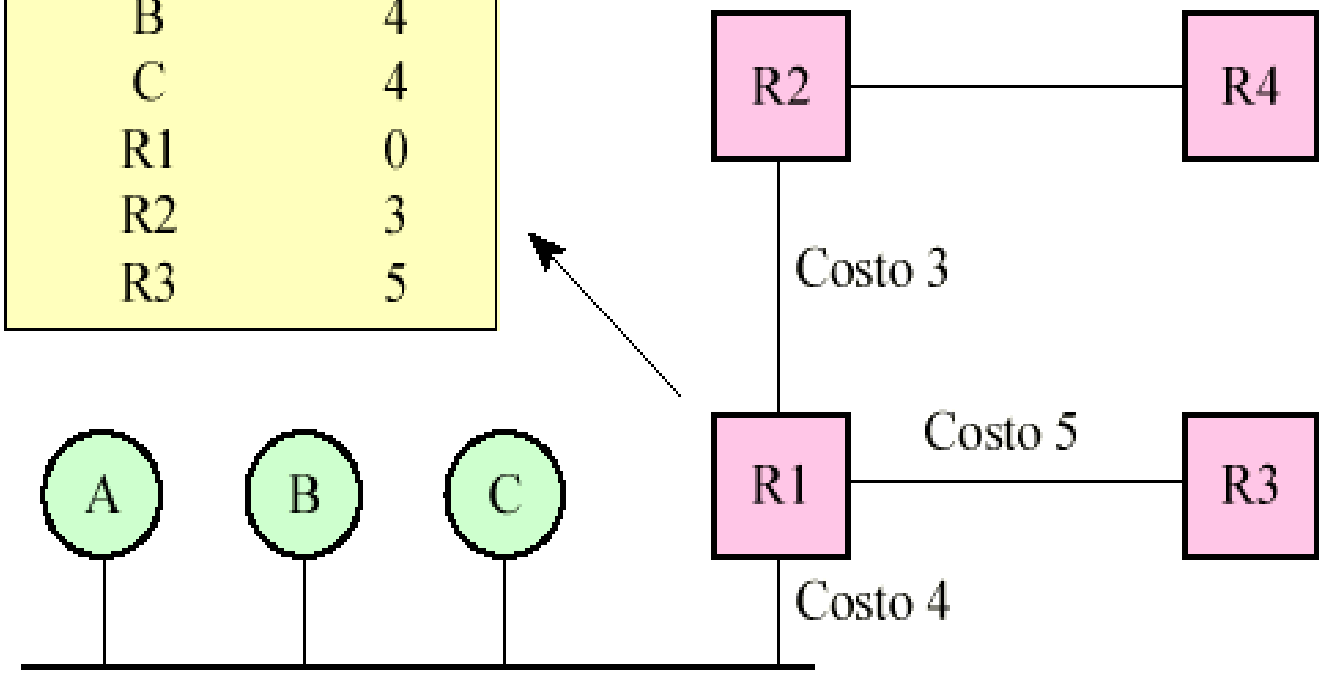
- Un LSP viene generato periodicamente, oppure quando viene rilevata una variazione nella topologia locale (adiacenze), ossia :
  - Viene riconosciuto un nuovo vicino
  - Il costo verso un vicino e' cambiato
  - Si e' persa la connettività verso un vicino precedentemente raggiungibile

- Un LSP è trasmesso in flooding su tutti i link del router
- I pacchetti LSP memorizzati nei router formano una mappa completa e aggiornata della rete:
  - *Link State Database*

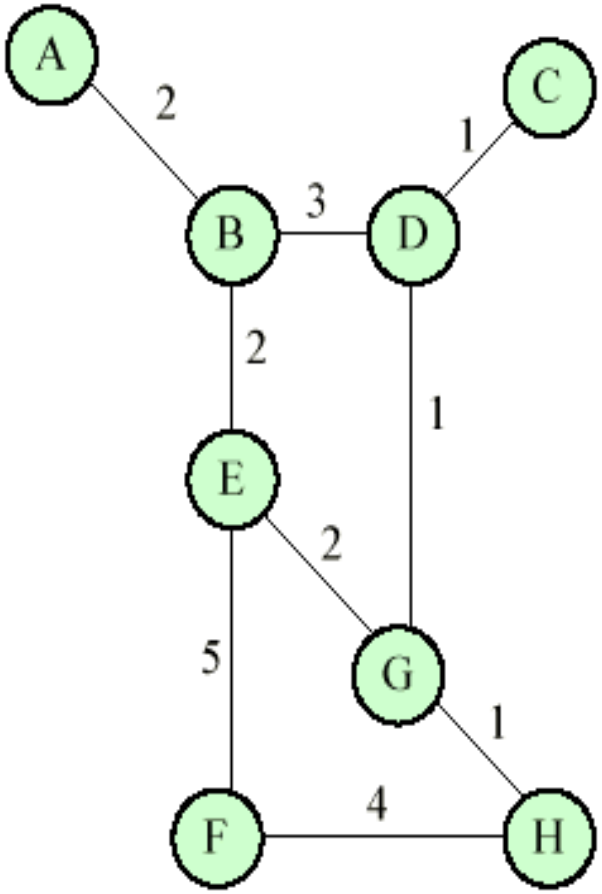
Esempio: trasmissione di un LSP

LSP trasmesso da R1

Adiacente	Costo
A	4
B	4
C	4
R1	0
R2	3
R3	5



Esempio: grafo della rete e LSP-DB



LSP Database

A	B/2		
B	A/2	D/3	E/2
C	D/1		
D	B/3	C/1	G/1
E	B/2	F/5	G/2
F	E/5	H/4	
G	D/1	E/2	H/1
H	F/4	G/1	

(replicato su ogni IS)

LSP database

SORGENTE

DESTINAZIONE

	A	B	C	D	E	F	G	H
A	0	2						
B	2	0		3	2			
C			0	1				
D		3	1	0			1	
E		2			0	5	2	
F					5	0		4
G				1	2		0	1
H						4	1	0

Questa rappresentazione è quella più appropriata per applicare l'agoritmo di Dijkstra

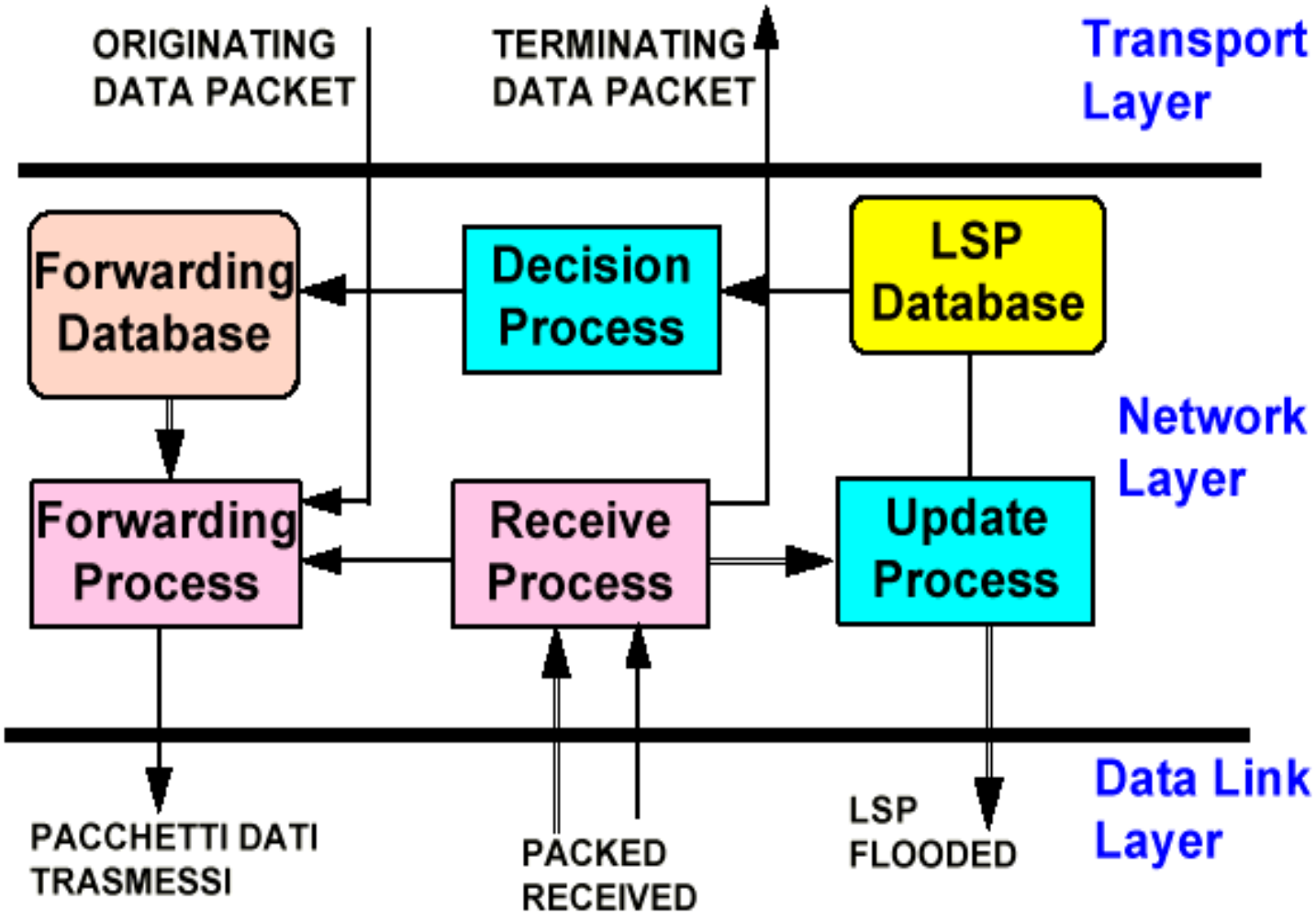
Gestione degli LSP

- All'atto della ricezione di un LSP, il router compie le seguenti azioni:
  1. se non ha mai ricevuto LSP da quel router o se l'LSP è più recente di quello precedentemente memorizzato:
     - memorizza il pacchetto
     - lo ritrasmette in flooding su tutte le linee eccetto quella da cui l'ha ricevuto
  2. se l'LSP ha lo stesso numero di sequenza di quello posseduto:
     - non fa nulla
  3. Se l'LSP è più vecchio di quello posseduto:
     - trasmette al mittente il pacchetto più recente

Routing: decisioni

- Il router elabora il *Link State Database* per produrre il Forwarding Database:
  - si pone come radice dello shortest-path tree
  - cerca lo shortest path per ogni nodo destinazione
  - memorizza il vicino (i vicini) che sono sullo shortest path verso ogni nodo destinazione
- Il *Forwarding Database* contiene, per ogni nodo destinazione:
  - l'insieme delle coppie {path, vicino}
  - la dimensione di tale insieme

Architettura di un router Link State



Link State: caratteristiche

- Vantaggi:
  - può gestire reti di grandi dimensioni
  - ha una convergenza rapida
  - difficilmente genera loop, e comunque è in grado di identificarli ed interromperli facilmente
  - facile da capire: ogni nodo ha la mappa della rete
- Svantaggi:
  - Molto complesso da realizzare:
    - Es: la prima implementazione ha richiesto alla Digital 5 anni

Esempio: tabelle di instradamento

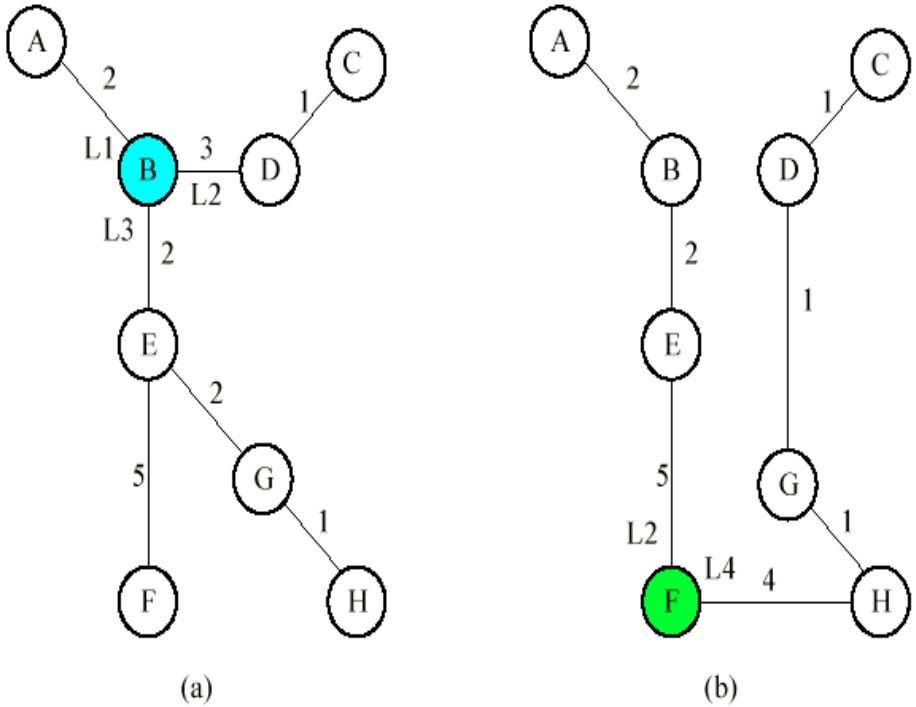


Tabella di B

A	L1
C	L2
D	L2
E	L3
F	L3
G	L3
H	L3

Tabella di F

A	L2
B	L2
C	L4
D	L4
E	L2
G	L4
H	L4

Algoritmo di Dijkstra

- Ogni nodo ha a disposizione il grafo della rete:
  - i nodi sono i router
  - gli archi sono le linee di collegamento tra router:
    - agli archi è associato un costo
- Ogni nodo usa l'algoritmo di Dijkstra per costruire lo *Shortest Path Tree* del grafo, ovvero l'albero dei cammini di costo minimo
- Ad ogni nodo si assegna un'etichetta che rappresenta il costo massimo per raggiungere quel nodo
- L'algoritmo modifica le etichette cercando di minimizzarne il valore e di renderle permanenti

Algoritmo di Dijkstra: formalizzazione

- La Topologia della rete è nota a tutti i nodi:
  - la diffusione è realizzata via "link state broadcast"
  - tutti i nodi hanno la stessa informazione
- Si calcola il percorso minimo da un nodo a tutti gli altri:
  - l'algoritmo fornisce la tavola di routing per quel nodo
- Iterativo: un nodo, dopo k iterazioni, conosce i cammini meno costosi verso k destinazioni

Notazione:

- c(i,j): costo collegamento da i a j: $c(i,j) \geq 0$
  - infinito se non c'è collegamento
  - per semplicità, $c(i,j) = c(j,i)$
- D(v): costo corrente del percorso, dalla sorgente al nodo v
- p(v): predecessore (collegato a v) lungo il cammino dalla sorgente a v
- N: insieme di nodi per cui la distanza è stata trovata

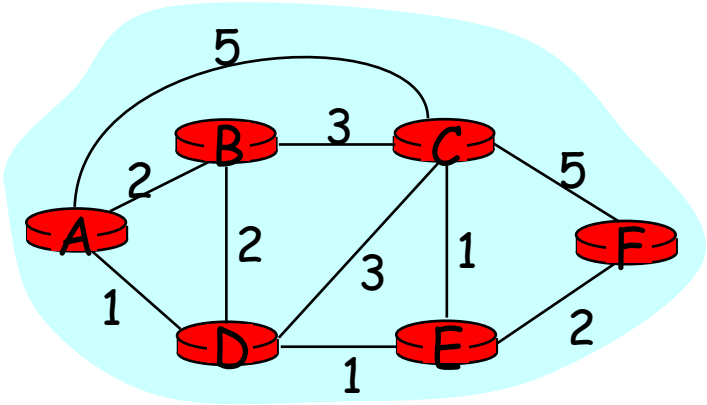
Algoritmo di Dijkstra (eseguito da A)

```
Inizializzazione:
  N = {A}
  per tutti i nodi  v
    if (v e'  adiacente a A)
      then D(v) = c(A,v)
      else D(v) = ∞

Loop
   sia w non in N tale che  D(w) è minimo
    aggiungi w a N
    aggiorna D(v) per ogni v adiacente a  w e non in N:
        D(v) = min( D(v), D(w) + c(w,v) )
    {il nuovo costo fino a v è o il vecchio costo, oppure il costo del
          cammino piu breve fino a w più il costo da w a v }
fino a quando tutti i nodi sono in N
```

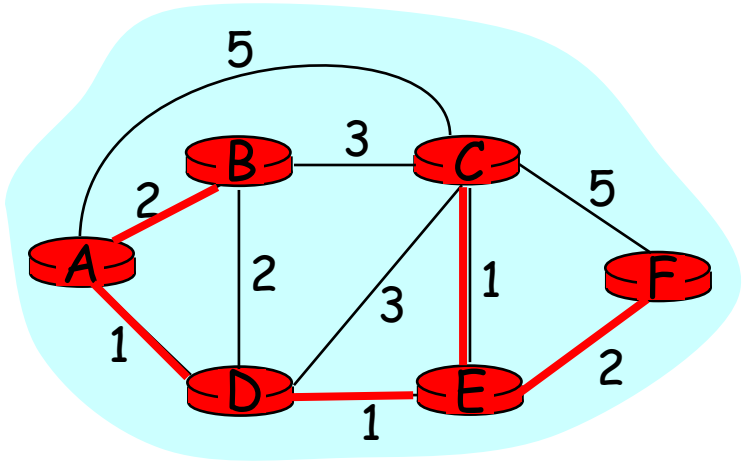
Algoritmo di Dijkstra: interpretazione

- L'algoritmo consiste in un passo di inizializzazione, più un ciclo di durata pari al numero di nodi della rete. Al termine avremo i percorsi più brevi dal nodo sorgente a tutti gli altri nodi
- Esempio. Calcoliamo sulla rete data i percorsi di costo minimo da A a tutte le possibili destinazioni. Ciascuna riga della tabella della slide seguente fornisce i valori delle variabili dell'algoritmo alla fine di ciascuna iterazione



Algoritmo di Dijkstra: esempio

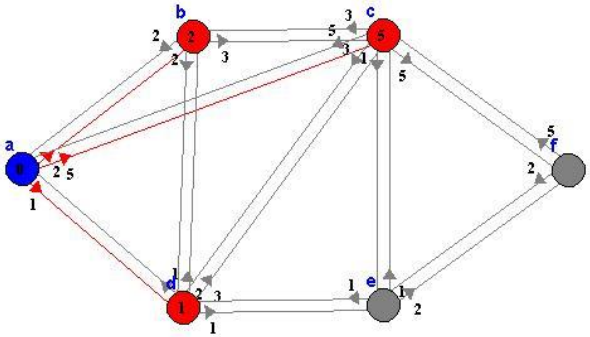
Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	infinity	infinity
1	AD	2,A	4,D		2,D	infinity
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					



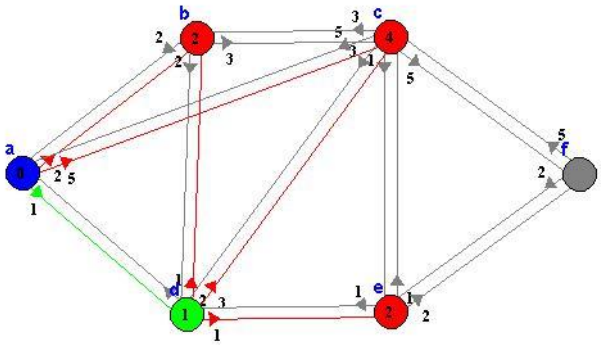
Notazione:

- c(i,j): costo collegamento da i a j (infinito se non c'e' collegamento e per semplicità c(i,j) = c(j,i))
- D(v): costo corrente del percorso, dalla sorgente al nodo v
- p(v): predecessore (collegato a v) lungo il cammino dalla sorgente a v
- N: insieme di nodi per cui la distanza è stata trovata

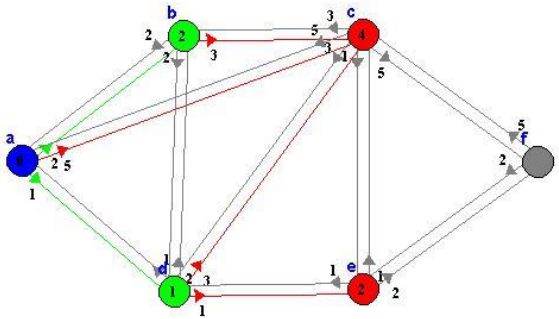
Dijkstra: esempio



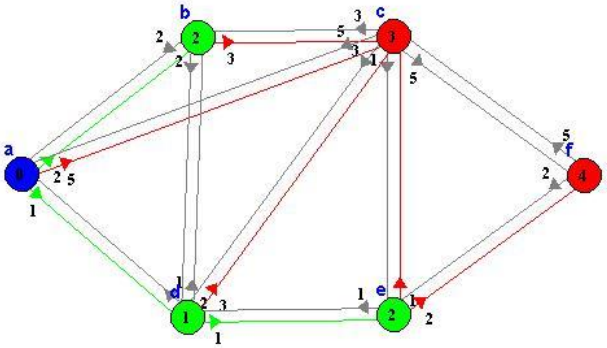
Passo 1



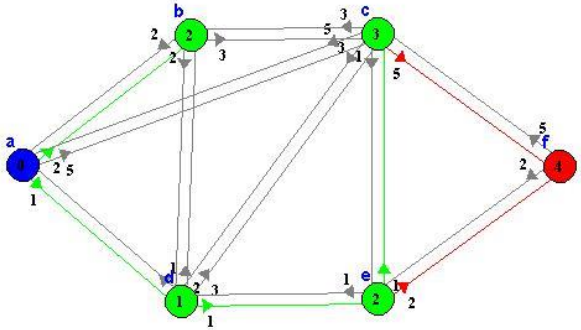
Passo 2



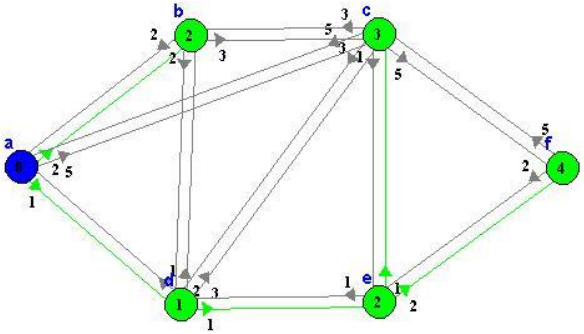
Passo 3



Passo 4

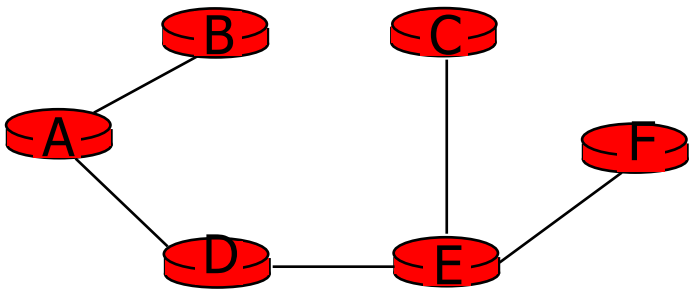


Passo 5



Passo 6

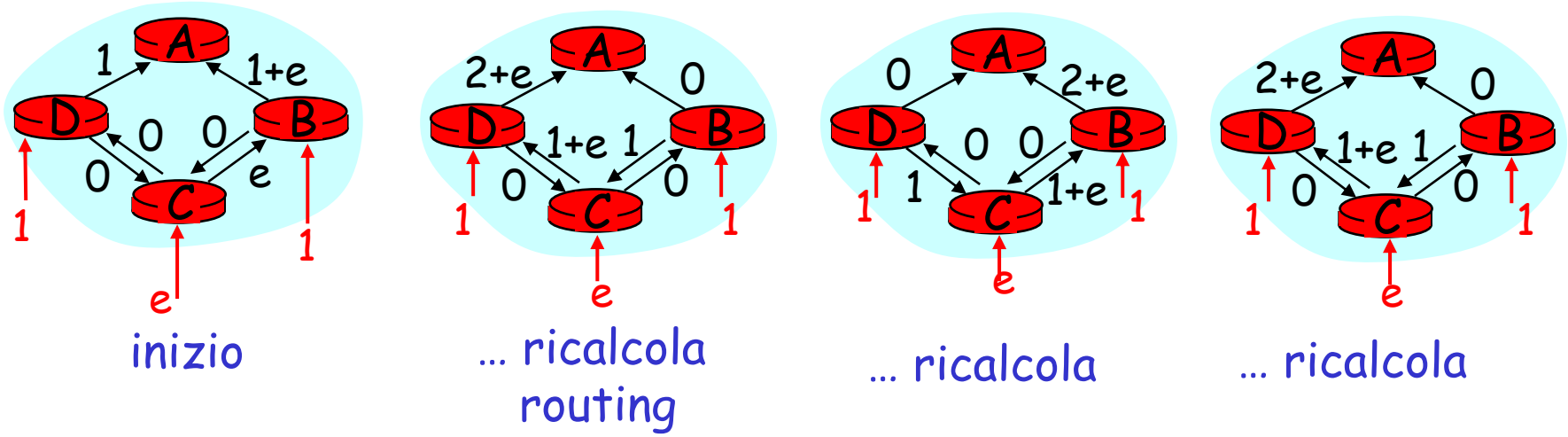
Esempio: tabella di instradamento in A



destination	link
B	(A,B)
C	(A,D)
D	(A,D)
E	(A,D)
F	(A,D)

Algoritmo di Dijkstra: discussione

Se il costo di un link è proporzionale al traffico su quel link, allora sono possibili oscillazioni



Soluzione: evitare la sincronizzazione nell'invio dei messaggi dei router