



Reti di Comunicazioni e Internet

Routing IP

Massimo Tornatore

Politecnico di Milano

Dip. di Elettronica, Informazione e Bioingegneria (DEIB)

Via Ponzio, 34/5 – 20133 Milan, Italy
massimo.tornatore @polimi.it

Sommario

- Algoritmi di instradamento
- Link-state/Dijkstra
- Instradamento nello strato IP

Riferimenti bibliografici: Pattavina 5.2 e 21.3

Algoritmi di instradamento

Introduzione

- **Requisiti** di un algoritmo di instradamento (*routing*)
 - Robustezza
 - Stabilità
 - Ottimalità
- **Quali pacchetti** necessitano di instradamento?
 - Pacchetti di segnalazione in servizi *circuito virtuale*
 - Pacchetti dati in servizi *datagram*
- **Chi decide** qual è la migliore soluzione di instradamento?
 - Algoritmi centralizzati, distribuiti o isolati

Algoritmi di instradamento

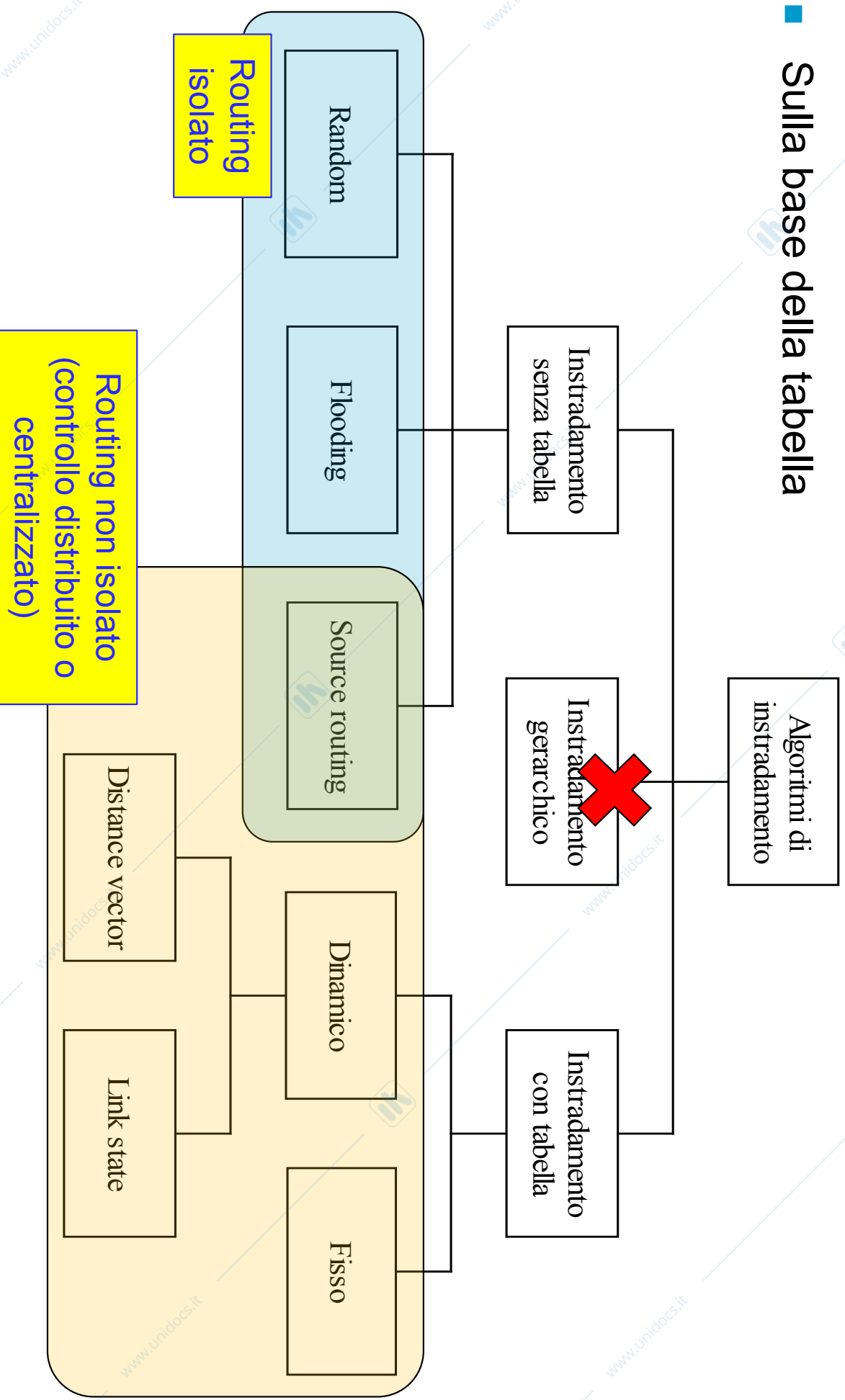
Tassonomia (1)

- Sulla base del nodo che opera la decisione
 - Algoritmi centralizzati
 - un unico centro di controllo prende tutte le decisioni
 - Algoritmi distribuiti
 - tutti i nodi cooperano per determinare il migliore instradamento
 - Algoritmi isolati:
 - singolo nodo seleziona intero percorso, tutti gli altri nodi solo transito (non prendono decisioni)
 - solo il nodo sorgente prende le proprie decisioni eventualmente anche in base a informazioni chieste ad altri nodi
 - tipicamente senza tabella
 - Comprende source routing con path discovery, random e flooding

Algoritmi di instradamento

Tassonomia (2)

■ Sulla base della tabella



Sommario

- Algoritmi di instradamento
 - Algoritmi di instradamento senza tabella
 - Algoritmi di instradamento con tabella
- Link-state/Dijkstra
- Instradamento nello strato IP

Algoritmi di instradamento senza tabella

Random

- Il link uscente è scelto casualmente
 - PRO: Molto robusto e a bassissima complessità
 - CON: Non ottimale e con tempi di ritardo con alta varianza
- Per avere equa **ripartizione del traffico** in rete:
 - Link in uscita scelto in base alla capacità disponibile sui link
 - Probabilità di selezionare link i :

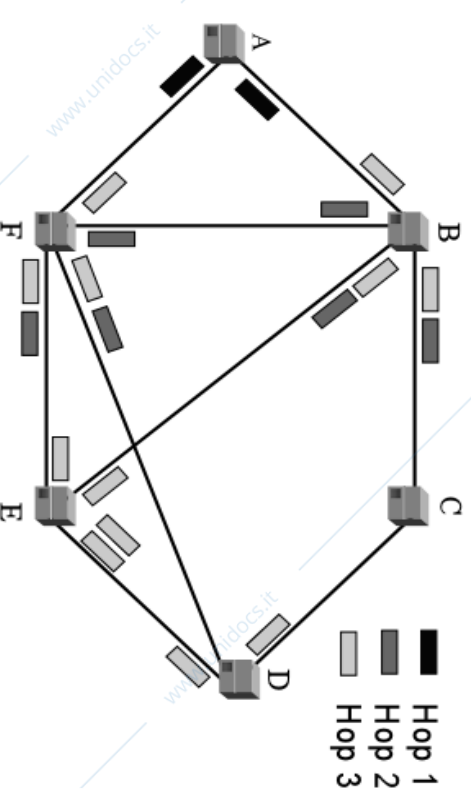
$$p_i = \frac{C_i}{\sum_{j=1}^H C_j}$$

- Variazione: Selective random
 - Instradamento deterministico quando la destinazione è adiacente
 - Richiede disponibilità di “tabella delle adiacenze” (non più completamente isolato)

Algoritmi di instradamento senza tabella

Flooding

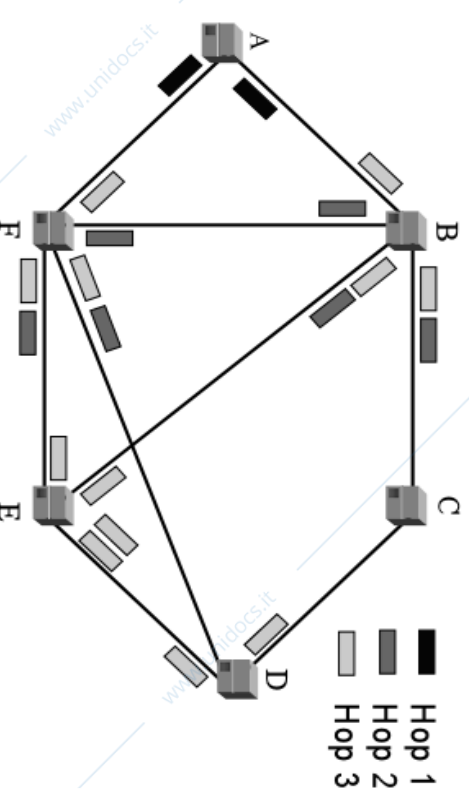
- Il pacchetto è replicato su tutti i link uscenti ad eccezione di quello di ingresso
 - Aumenta il traffico interno a parità di quello esterno
 - Nota: si può limitare il traffico inoltrato, se ogni nodo
 - Scarta i pacchetti già transitati oppure
 - Scarta i pacchetti il cui massimo numero di salti si è esaurito
 - Necessario un contatore nel pacchetto (as es. il TTL)
 - Se si usa l'info sul "diametro" della rete (distanza massima tra nodi) si riduce ulteriormente il traffico in rete
 - Robustissimo (adottato in applicazioni militari)
- Esempio (sorgente A, destinazione C)
 - NB: nell'esempio non c'è scarto



Algoritmi di instradamento senza tabella

Flooding

- Il pacchetto è replicato su tutti i link uscenti ad eccezione di quello di ingresso
 - Aumenta il traffico interno a parità di quello esterno
 - Nota: si può limitare il traffico inoltrato, se ogni nodo
 - Scarta i pacchetti già transitati oppure
 - Scarta i pacchetti il cui massimo numero di salti si è esaurito
 - Necessario un contatore nel pacchetto (as es. il TTL)
 - Se si usa l'info sul "diametro" della rete (distanza massima tra nodi) si riduce ulteriormente il traffico in rete
 - Robustissimo (adottato in applicazioni militari)
- Esempio (sorgente A, destinazione C)
 - NB: nell'esempio non c'è scarto
- Variazione: Selective flooding
 - Instradamento solo su un gruppo di linee uscenti (uso di una "tabella" ridotta con link diretti verso la destinazione)



Algoritmi di instradamento senza tabella

Source routing

- Il nodo sorgente predetermina la strada scrivendola nel pacchetto
 - Sequenza di interfacce da attraversare: $H_s, N_1, N_2, \dots, N_L, H_d$ (N_i identifica l'indirizzo dell'interfaccia al nodo i)
- Path determination
 - Fornita da un path server (soluzione centralizzata, semplice ma inaffidabile)
 - Fornita da un path discovery (soluzione isolata)
 - Il nodo sorgente invia in flooding un pacchetto alla destinazione
 - Ogni nodo attraversato riempie il primo campo N_i ancora vuoto
 - Il nodo destinazione sceglie tra i pacchetti arrivati quello che identifica la via migliore (primo arrivato o quello con il minore numero di nodi attraversati)
 - Il nodo destinazione comunica la sua scelta al nodo sorgente che utilizza la via scelta

Sommario

- Algoritmi di instradamento dello strato di rete
 - Algoritmi di instradamento senza tabella
 - Algoritmi di instradamento con tabella
- Instradamento nello strato IP
- Distance vector
- Link state

Algoritmi di instradamento con tabella

*quello di IP!

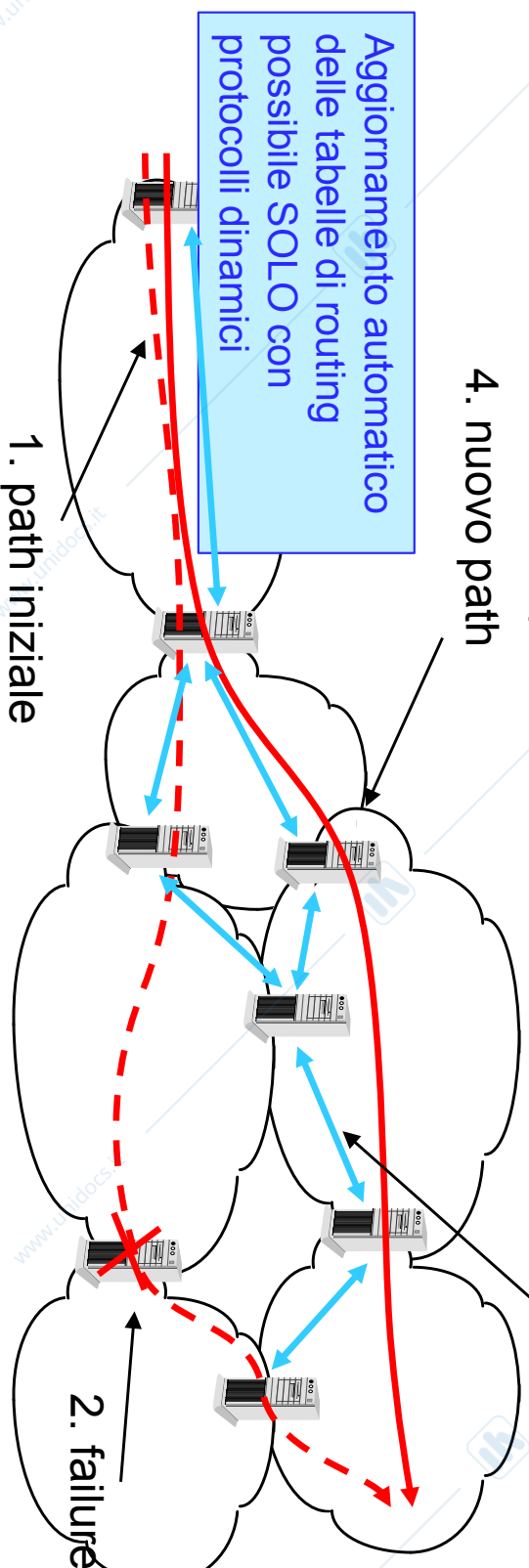
- Basati su instradamento a **distanza minima** (o **costo minimo**)
- Richiede la definizione di una **metrica di costo**. Ad esempio:
 - Ritardo dei link
 - Numero di link attraversati (salti o «hop»)
 - Numero medio di pacchetti in coda sull'intero percorso
 - ecc.
- Le tabelle di instradamento indicano per ogni destinazione di rete il **nodo successivo** verso cui instradare il pacchetto
- Due possibili implementazioni: **instradamento statico** o **dinamico**

Routing statico

- Forma più semplice di routing IP
- Tabelle di routing compilate offline, per esempio **manualmente** dall'amministratore di rete
 - Fase di design
 - Per ciascun router viene selezionata una rotta per ciascuna possibile altra sottorete
 - Fase di configurazione
 - Le tabelle di routing sono configurate tramite opportune istruzioni
- *Quando è opportuno utilizzarlo?*
 - Periferia della rete
 - Sistemi (ad es. ISP) con bassa connettività (poche righe da compilare manualmente)
 - Necessità di fare *load-balancing*

Routing dinamico (adattativo)

- I protocolli di routing in IP (OSPF, RIP) sono **dinamici**
- Le entry delle tabelle di routing cambiano nel tempo, in base a:
 - **Guasti** nei link (*failures*)
 - Quando un link si interrompe è necessario aggirarlo
 - Cambiamenti nella **topologia** di rete
 - Quando viene aggiunto un nuovo router backbone, considera il suo utilizzo
 - **Carico** di rete e **congestione**
 - Quando un link è poco usato, conviene sfruttarlo



Sommario

- Algoritmi di instradamento dello strato di rete
- Distance vector
 - Non si fa!
- Link state/Dijkstra
- Instradamento nello strato IP

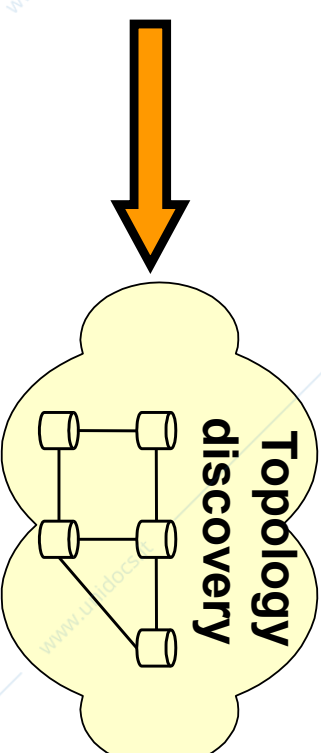
LSP Dissemination e Topology discovery

LSP che arrivano a C

- [A: AB,AE]
- [B: AB,BD,BC]
- [D: BD,CD,DE]
- [E: AE,DE]

Router C :

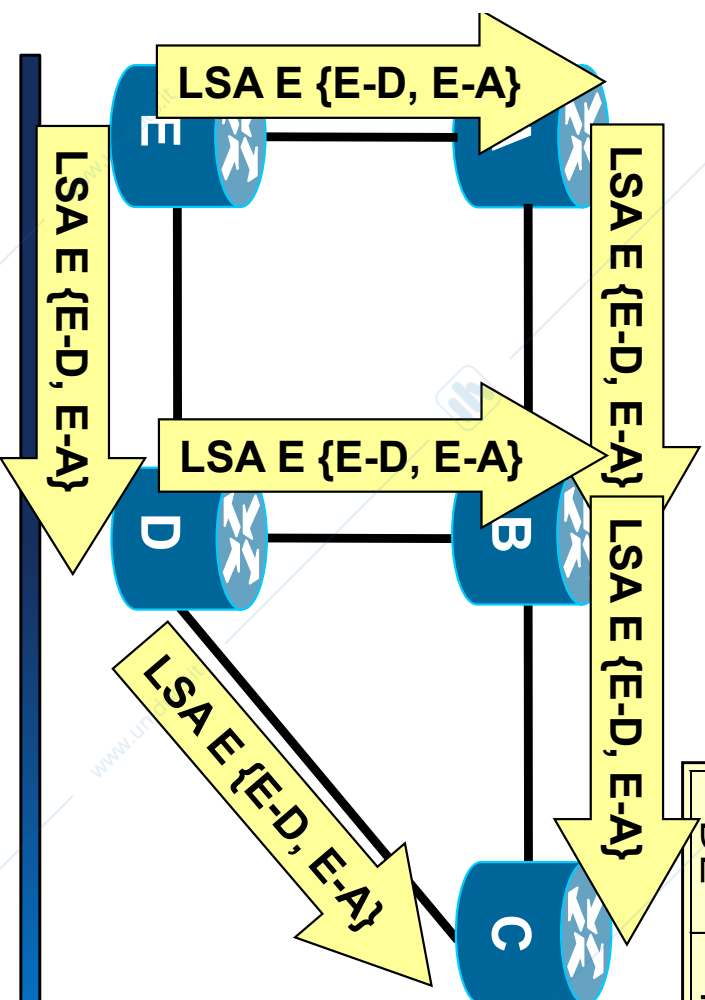
Link	Cost	From
AB	1	A
AE	1	A
BD	1	B
BC	1	B
CD	1	D
DE	1	D



Dijkstra :
cammini minimi

Router C :

Dest.	Next hop	Interface
A	B	BC
B	Direct	BC
D	Direct	CD
E	D	CD



Algoritmo link state

- Ogni nodo misura il **costo dei collegamenti** (ad esempio secondo la metrica “distanza” o “ritardo”) verso tutti i suoi vicini
 - Invia un vettore di “**link state**”
 - Link state funziona solo con costi di **segno positivo**
- Questa distanza è comunicata a tutti gli altri nodi con tecnica **flooding**
- Ogni nodo ha **visibilità completa della rete** e può così costruire i percorsi a minima distanza verso ciascun altro nodo
- I costi dei link possono variare
 - serve inviare **nuovi link state** all’occorrenza
- Anche se non ci sono aggiornamenti i link state vengono comunque inviati periodicamente
 - periodo di aggiornamento nell’ordine dei 10 s

Algoritmo di Dijkstra:

Instradamento a minima distanza o «cammino minimo»

- Si applica al generico nodo sorgente s

N = nodi della rete

M = insieme dei nodi del MST corrente

$V(M)$ = nodi "vicini" all'insieme M

d_{ij} = costo della via diretta da i a j ($d_{ij} = \infty$ in assenza del link $i - j$)

D_j = costo della via a minimo costo da s a j

P_j = nodo predecessore del nodo j

1 $M = \{s\}$

$D_j = d_{sj}$ $\forall j \in N - \{s\}$, $D_s = 0$

$P_j = s$ $\forall j \in N - \{s\}$

2 **Select** $k \in V \setminus M$ $D_k = \min_{i \in V \setminus M} D_i$

$M = M \cup \{k\}$, connect k to P_k in MST

if $D_j > D_k + d_{kj}$ then $P_j = k$ $\forall j \in V(M)$

$D_j = \min\{D_j, D_k + d_{kj}\}$ $\forall j \in V(M)$

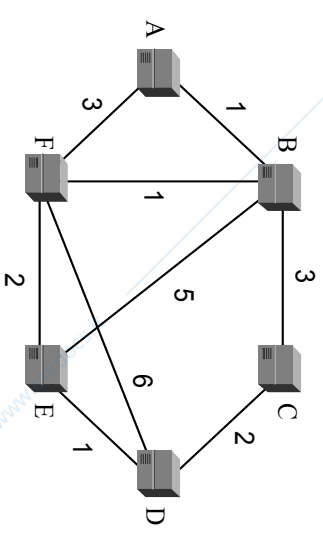
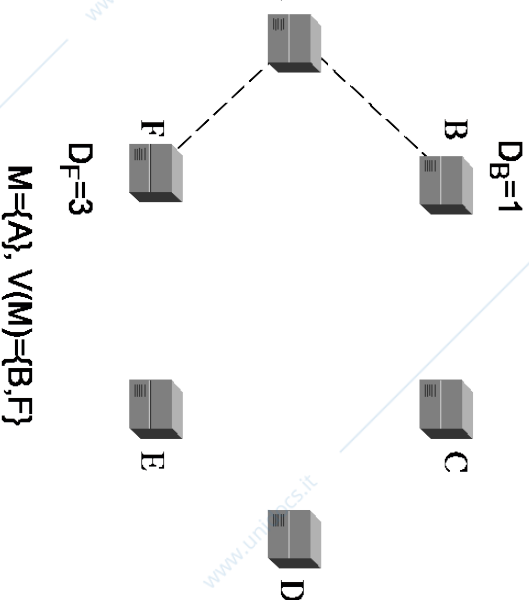
3 If $M = N$ stop

else go to 2

Algoritmo di Dijkstra

Esempio

$\text{Select } k \in V \setminus \{M\} \mid D_k = \min_{i \in V \setminus \{M\}} D_i$
 $M = M + \{k\}$; connect k to P_k in MST A $\forall j \in V(M)$
 If $D_j > D_k + d_{kj} \Rightarrow P_j = k$ $\forall j \in V(M)$
 $D_j = \min \{D_j, D_k + d_{kj}\} \forall j \in V(M)$

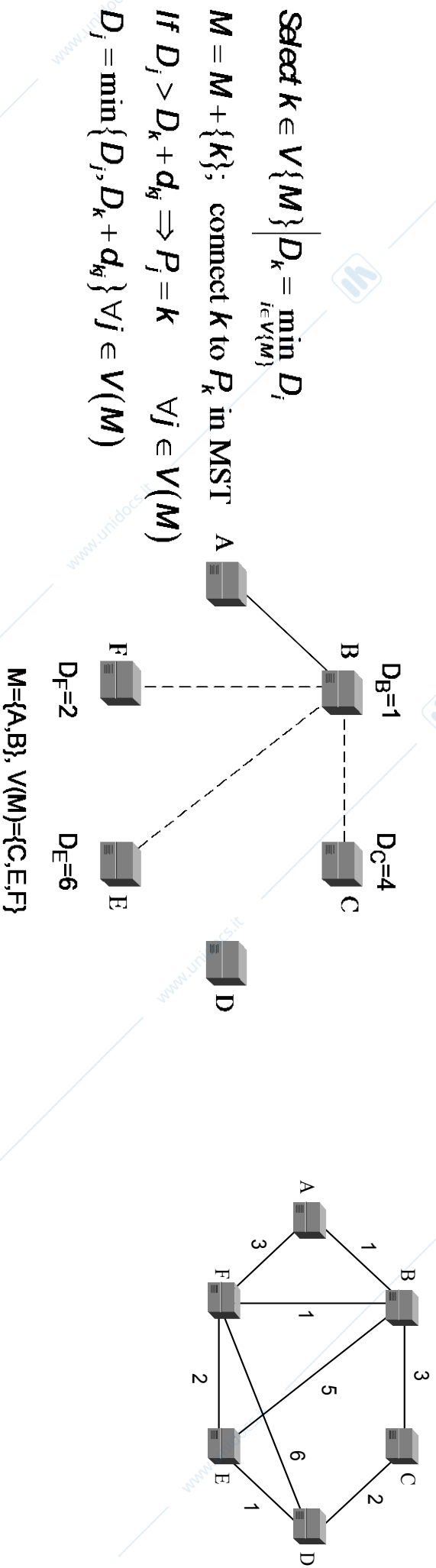


Dest.	M={A}	
	Cost	Route
A	-	
B	1	B
C	∞	
D	∞	
E	∞	
F	3	F

Tabella di instradamento del nodo A

Algoritmo di Dijkstra

Esempio



Dest.	M={A}		M={A,B}	
	Cost	Route	Cost	Route
A	-		-	
B	1	B	1	B
C	∞		4	B
D	∞		∞	
E	∞		6	B
F	3	F	2	B

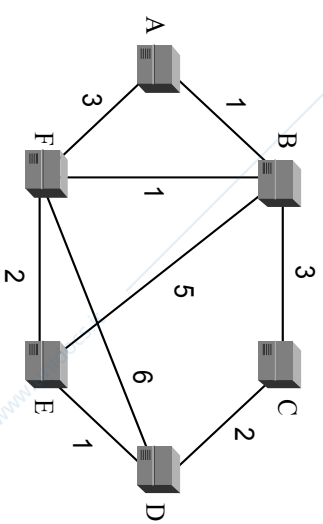
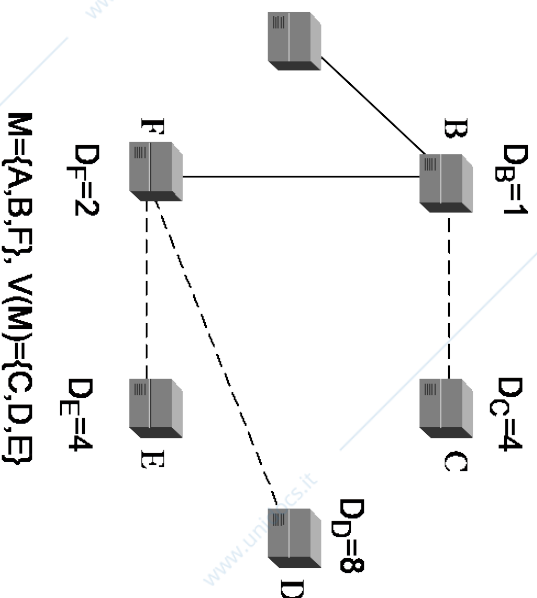
Tabella di instradamento del nodo A

Algoritmo di Dijkstra

Esempio

$$D_k = \min_{i \in V(M)} D_i$$

Select $k \in V(M)$ | $D_k = \min_{i \in V(M)} D_i$
 $M = M + \{k\}$; connect k to P_k in MST A
If $D_j > D_k + d_{kj} \Rightarrow P_j = k \quad \forall j \in V(M)$
 $D_j = \min \{D_j, D_k + d_{kj}\} \forall j \in V(M)$



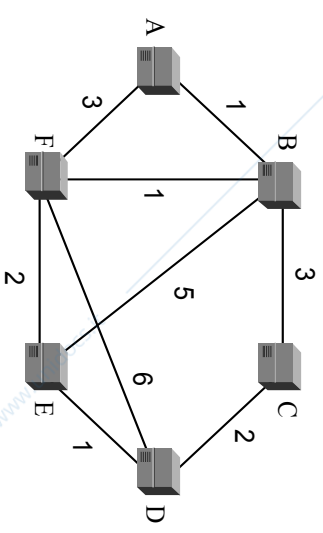
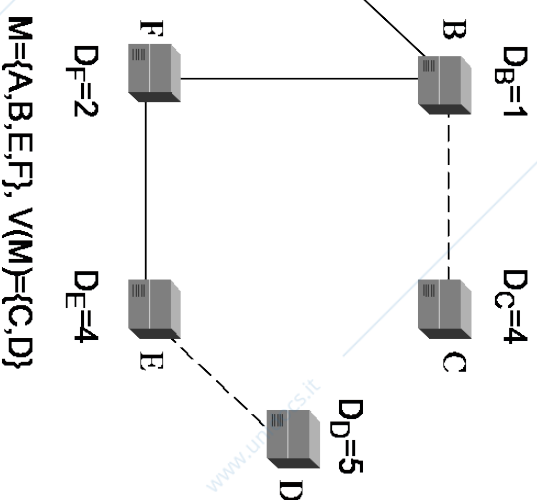
Dest.	M={A}		M={A,B}		M={A,B,F}	
	Cost	Route	Cost	Route	Cost	Route
A	-		-		-	
B	1	B	1	B	1	B
C	∞		4	B	4	B
D	∞		∞		8	B
E	∞		6	B	4	B
F	3	F	2	B	2	B

Tabella di instradamento del nodo A

Algoritmo di Dijkstra

Esempio

$M = M + \{k\}$; connect k to P_k in MST
 If $D_j > D_k + d_{kj} \Rightarrow P_j = k \quad \forall j \in V(M)$
 $D_j = \min \{D_j, D_k + d_{kj}\} \forall j \in V(M)$



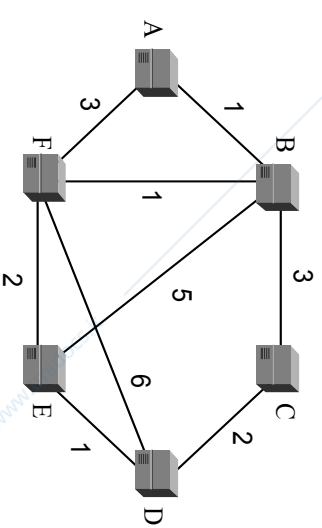
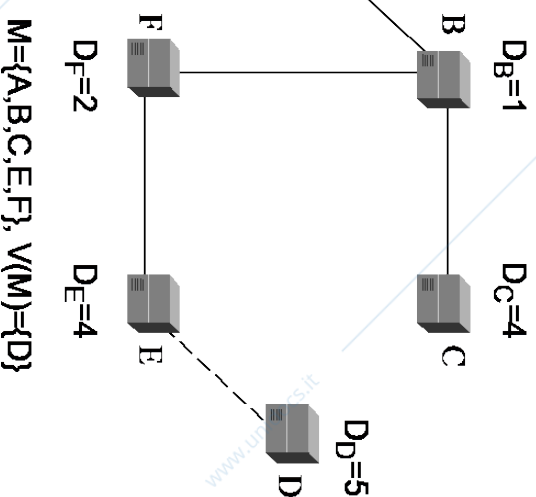
Dest.	M={A}		M={A,B}		M={A,B,F}		M={A,B,E,F}	
	Cost	Route	Cost	Route	Cost	Route	Cost	Route
A	-		-		-		-	
B	1	B	1	B	1	B	1	B
C	∞		4	B	4	B	4	B
D	∞		∞		8	B	5	B
E	∞		6	B	4	B	4	B
F	3	F	2	B	2	B	2	B

Tabella di instradamento del nodo A

Algoritmo di Dijkstra

Esempio

$D_k = \min_{i \in V(M)} D_i$
Select $k \in V(M)$ | $D_k = \min_{i \in V(M)} D_i$
 $M = M + \{k\}$; connect k to P_k in MST A
If $D_j > D_k + d_{kj} \Rightarrow P_j = k \quad \forall j \in V(M)$
 $D_j = \min \{D_j, D_k + d_{kj}\} \forall j \in V(M)$

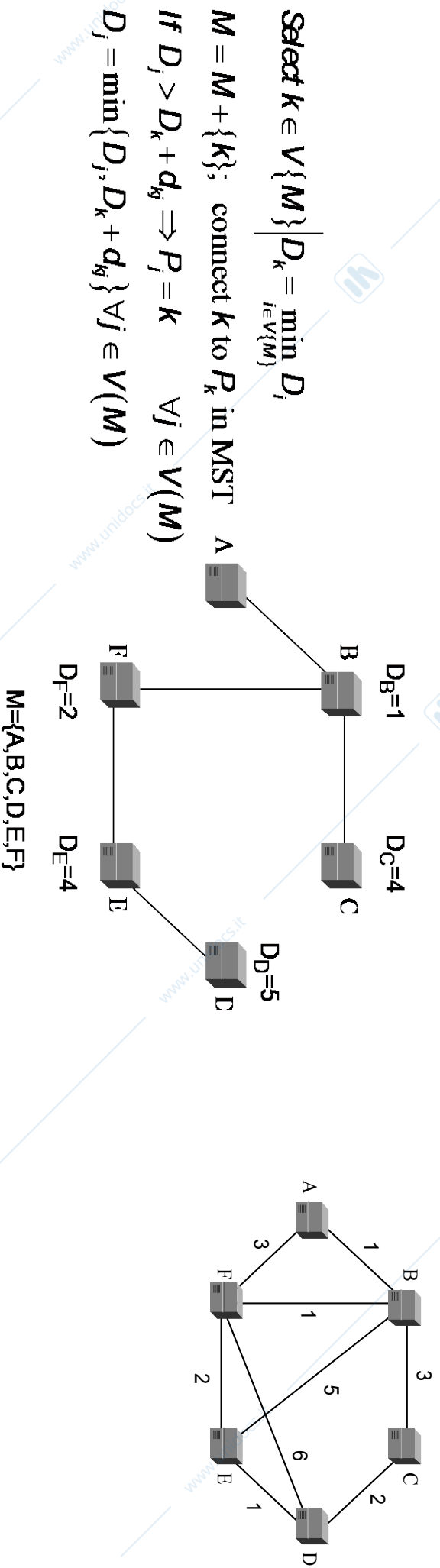


Dest.	M={A}	M={A,B}	M={A,B,F}	M={A,B,E,F}	M={A,B,C,E,F}	
	Cost	Route	Cost	Route	Cost	Route
A	-		-		-	
B	1	B	1	B	1	B
C	∞		4	B	4	B
D	∞		8	B	5	B
E	∞		4	B	4	B
F	3	F	2	B	2	B

Tabella di instradamento del nodo A

Algoritmo di Dijkstra

Esempio

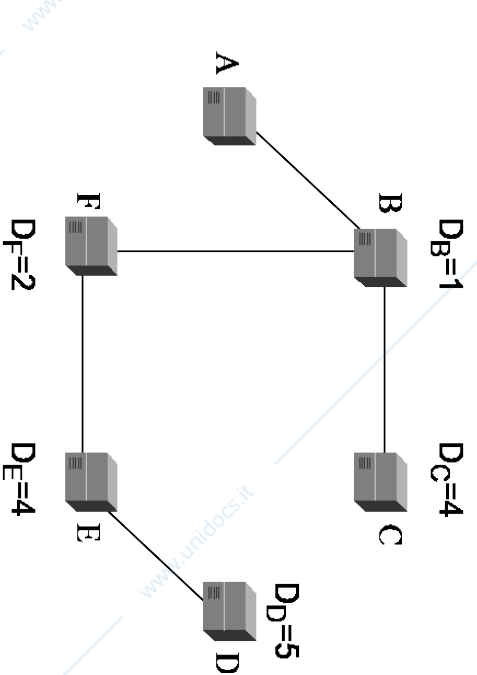
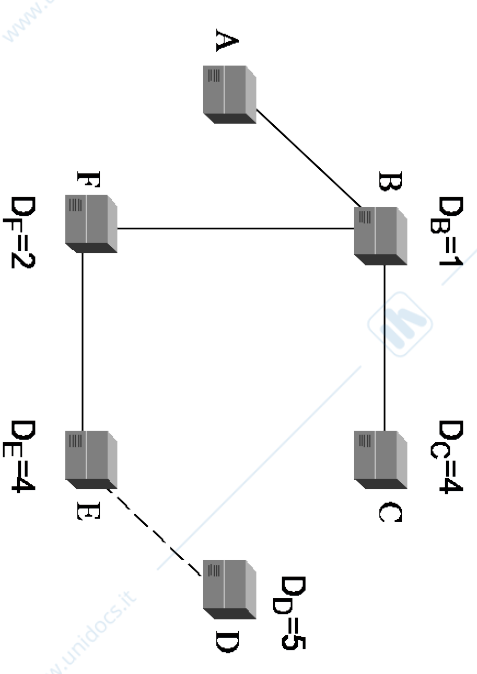
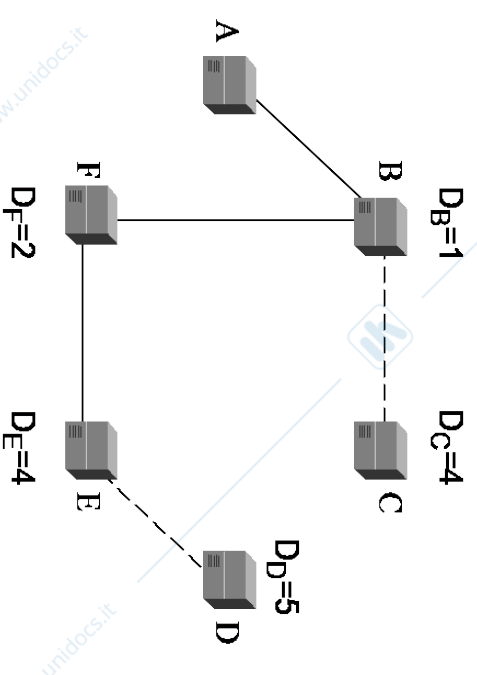
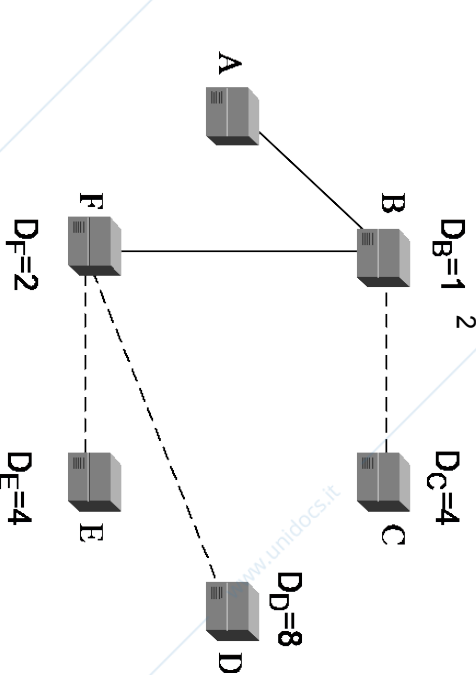
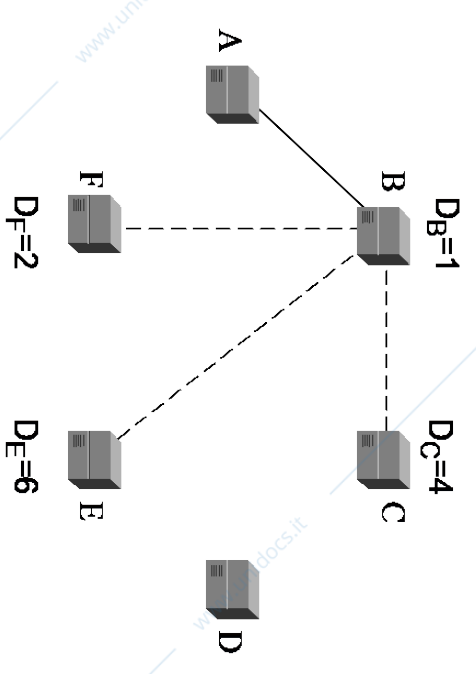
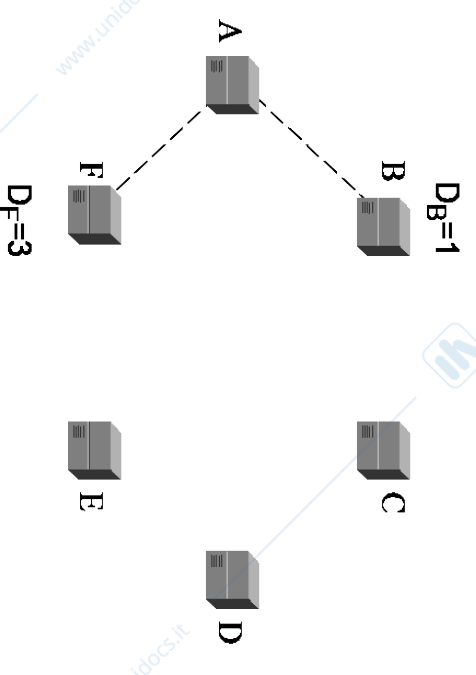
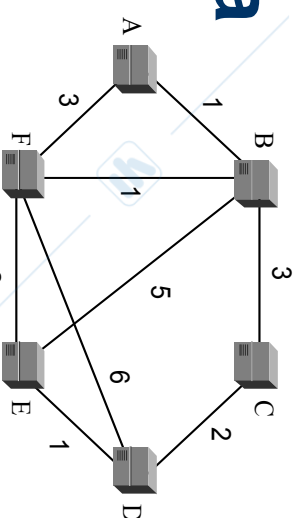


Dest.	M={A}	M={A,B}	M={A,B,F}	M={A,B,E,F}	M={A,B,C,E,F}	M={A,B,C,D,E,F}	
Cost	Route	Cost	Route	Cost	Route	Cost	Route
A	-	-	-	-	-	-	
B	1	B	1	B	1	B	
C	∞	4	B	4	B	4	
D	∞	∞	4	B	4	B	
E	∞	6	B	4	B	4	
F	3	F	2	B	2	B	

Tabella di instradamento del nodo A

Algoritmo di Dijkstra

Esempio



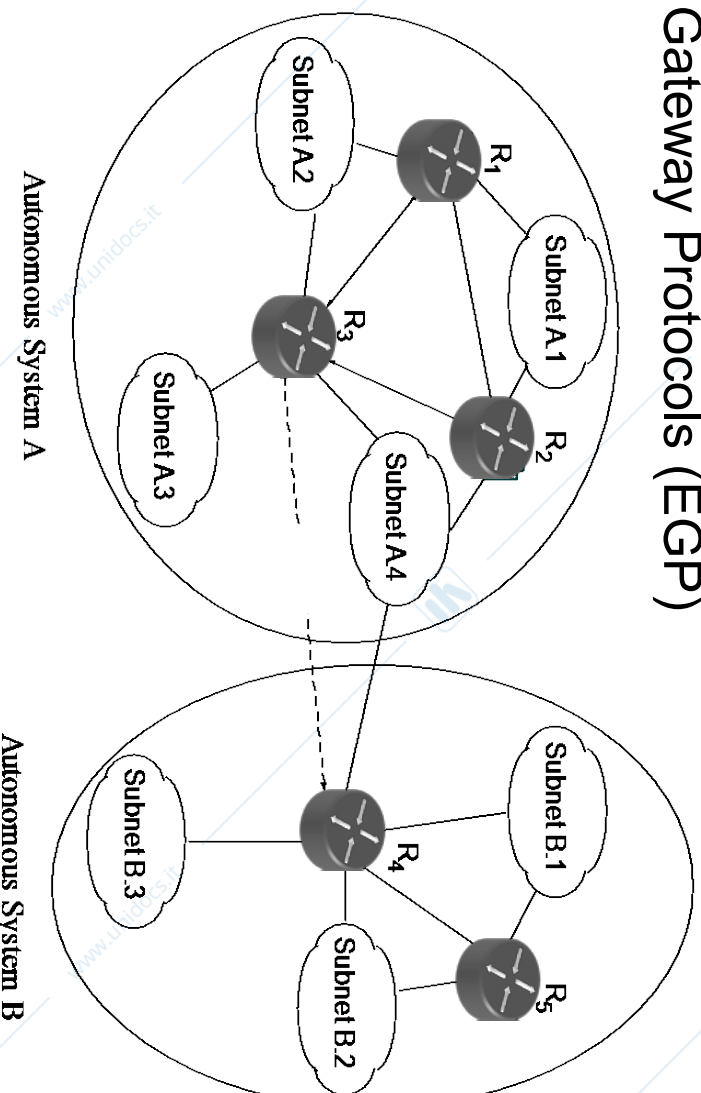
Sommario

- Algoritmi di instradamento dello strato di rete
- Link State/Dijkstra
- Instradamento nello strato IP

- Vediamo ora come i concetti generali introdotti nella sezione precedente si applicano nel protocollo IP
- Fondamentalmente si usa:
 - Instradamento dinamico con tabella mediante link state (algoritmo di Dijkstra)

Autonomous system, interior / exterior protocols

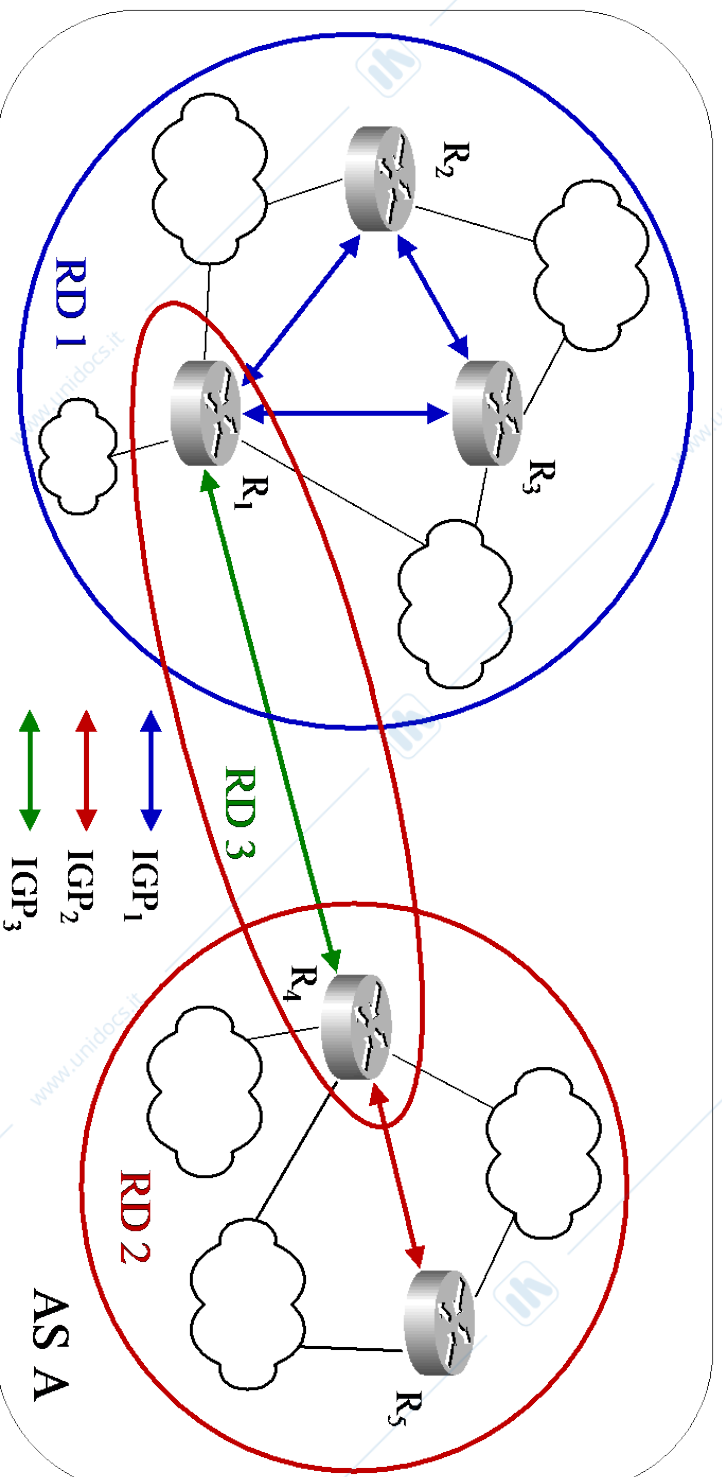
- Non è pensabile applicare LS/Dijkstra a tutta la rete Internet
 - Suddivisione in Autonomous Systems (AS)
- AS: insieme di sottoreti connesse da router omogenei, possibilmente gestiti da un'unica entità
- 2 categorie generali di protocolli di routing
 - Interior Gateway Protocols (IGP)
 - Exterior Gateway Protocols (EGP)



Routing IP

Autonomous system vs. routing domain

- Gli AS non usano necessariamente un unico interior gateway protocol
 - Routing Domain (RD) corrisponde ad una porzione dell'AS dove si usa un solo protocollo di routing IGP
 - Router che appartengono a diversi RDs possono usare protocolli di routing differenti



Protocolli di Routing: alcuni esempi

- IGP – Intra-domain
Distance vector
 - RIP: Routing Information Protocol, version 1 and 2
 - IGRP: Interior Gateway Routing Protocol (proprietary CISCO)
 - IS-IS: Intermediate System to Intermediate System
 - OSPF: Open Shortest Path FirstLink state
- EGP – Inter-domain
 - BGP: Border Gateway Protocol Path vector
 - EGP (RFC827 and RFC 924): now obsolete
 - Same name as whole class!

Protocollo Open Shortest Path First (OSPF)

Concetti base

- Open Shortest Path First
 - Open = non-proprietario; SPF = algoritmo di Dijkstra : RFC 1583
 - v2 RFC 2328 (sostituisce RFC 2178)
- Ogni router deve
 - “Scoprire” i suoi vicini
 - Misurare il costo per raggiungere ciascun vicino
 - Costruire un pacchetto (Link State Packet (LSP)) che contenga le informazioni su tutti i link connessi
 - Inviare il pacchetto a tutti gli altri router (non solo ai vicini), tipicamente con tecnica *flooding* (*piccolo pacchetto! Solo l'informazione dei vicini*)
- Algoritmo di routing link state
 - Si adotta routing shortest path (Dijkstra)
 - Ogni router ha cognizione dell'intera rete e può calcolare il MST
- Si inviano aggiornamenti al seguito di **eventi di modifica dei costi** dei link (*event update*)
 - Non è necessario inviare update periodici

IP & OSPF

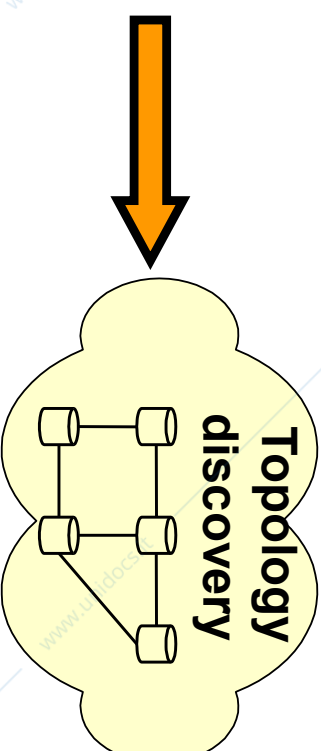
LSF dissemination and Topology discovery

LSP che arrivano a C

- [A: AB,AE]
- [B: AB,BD,BC]
- [D: BD,CD,DE]
- [E: AE,DE]

Router C :

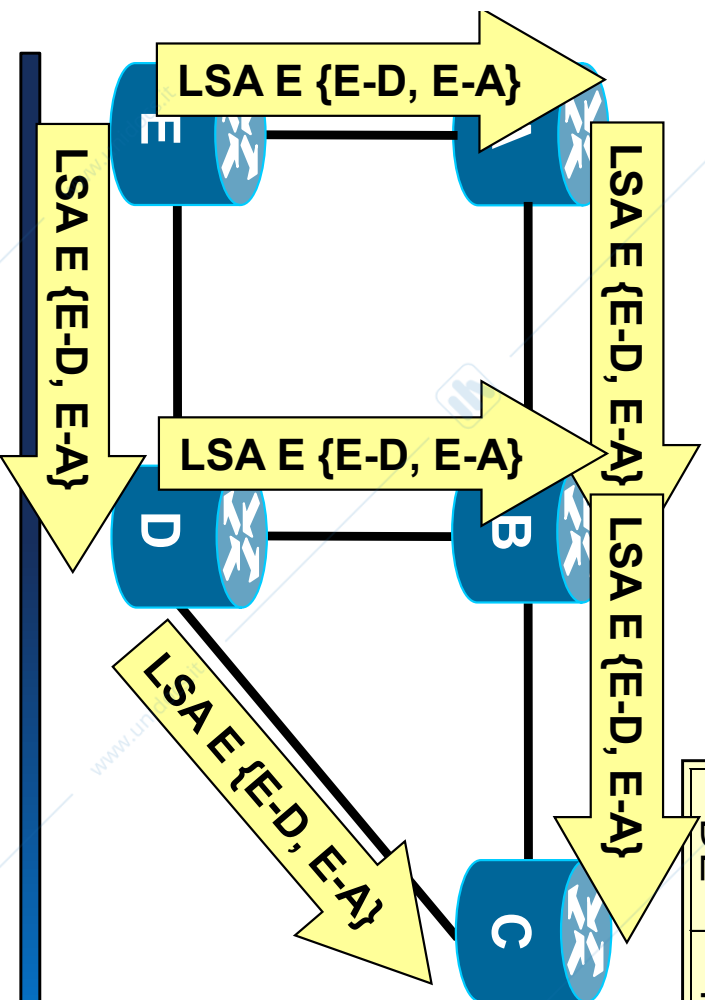
Link	Cost	From
AB	1	A
AE	1	A
BD	1	B
BC	1	B
CD	1	D
DE	1	D



Dijkstra :
cammini minimi

Router C :

Dest.	Next hop	Interface
A	B	BC
B	Direct	BC
D	Direct	CD
E	D	CD



Protocollo Open Shortest Path First (OSPF)

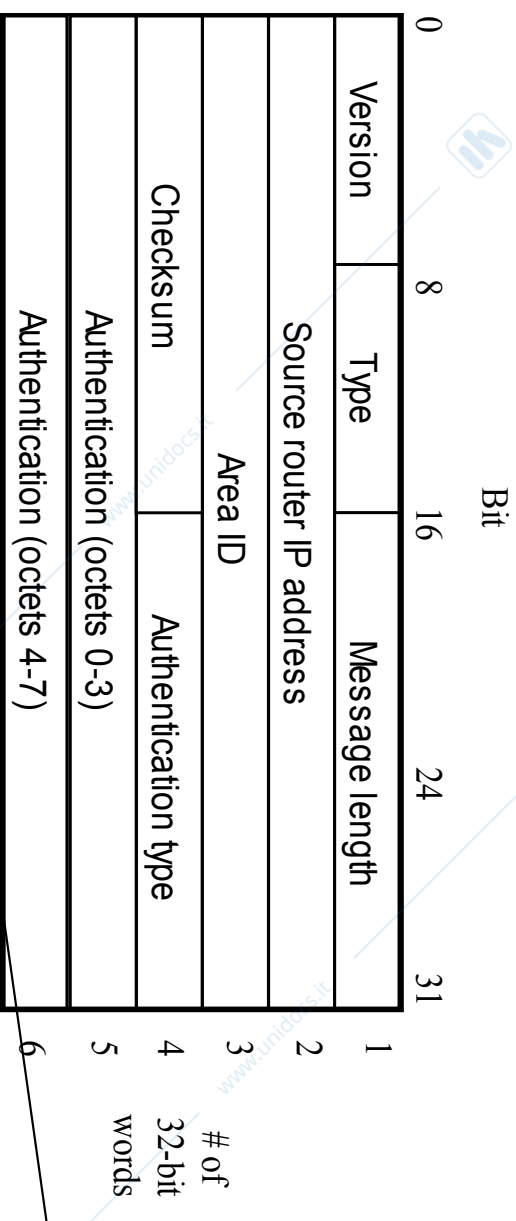
Caratteristiche (I)

- I messaggi OSPF sono trasportati nel payload dei pacchetti IP
- Aggiornamento delle tabelle di routing
 - Si effettua tramite selective flooding
 - **Periodico**: LSP sono inviati in broadcast dal router per segnalare eventi di up/down di link ad esso afferenti ribadendo il costo dei link
 - **Non-periodico**: in seguito a cambiamenti nel costo dei link
- Routing basato sulla tipologia di servizio
 - Uso del campo Type of service (TOS) del datagramma IP
 - 16 metriche, una per ogni valore che può assumere il campo TOS (flag D/T/R/C)
 - Metriche di distanza: una qualunque combinazione di ritardo, costo, bit-rate, etc.
 - Configurata dall'amministratore di rete

Formato dei pacchetti OSPF

Link State Packets (LSP)

- Header dei messaggi (comune a tutti i tipi di messaggi)



Authentication. Può contenere password (in questo caso il contenuto è crittografato)

Il payload contiene un insieme di Link State Updates (LSU), uno per ciascun link di cui inviare informazioni

- Tipi di messaggi

- **Hello** Testa la raggiungibilità dei vicini
- **Link state request (LSR)** Richiede informazioni per uno specifico link
- **Link state update (LSU)** Fornisce il costo dei link vicini al sender
- **Link state ack (LSA)** Riscontra l'update (LSU)
- **Database description** Annuncia quali update possied il sender

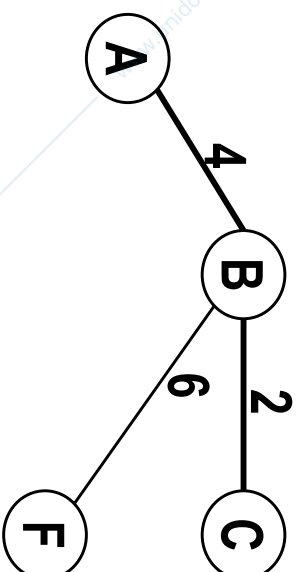
OSPF

Esempio - 1

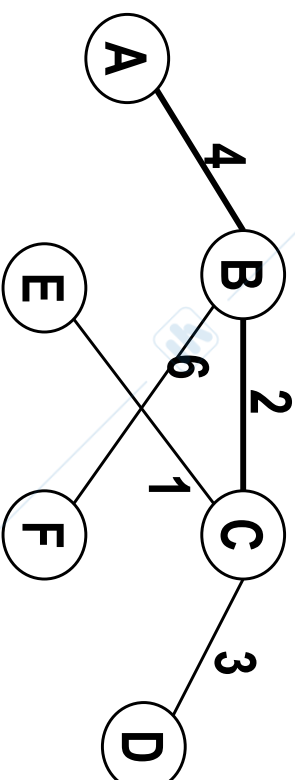
Pacchetti ricevuti

Network discovery

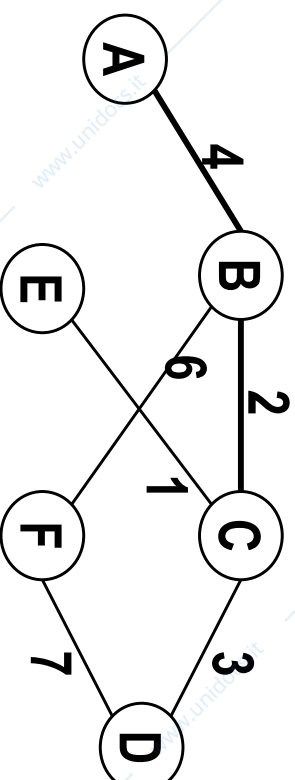
B	
A	4
C	2
F	6



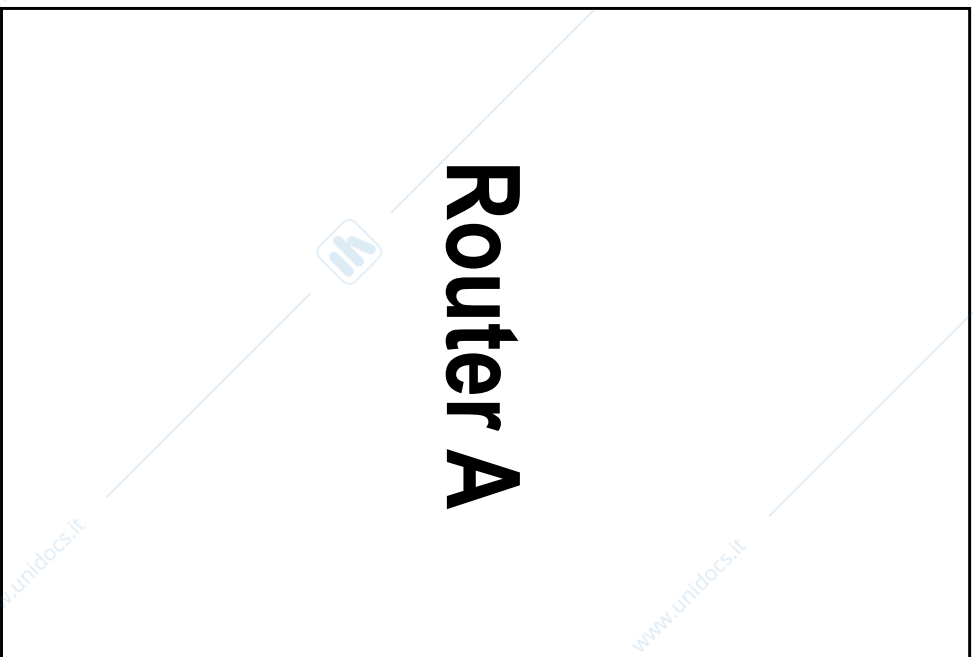
C	
B	2
D	3
E	1



D	
C	3
F	7



Router A



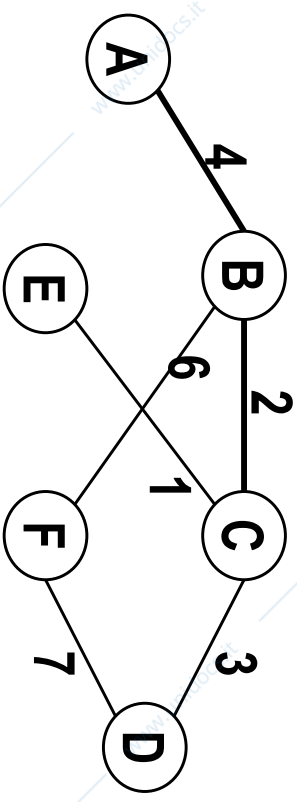
OSPF

Esempio - 2

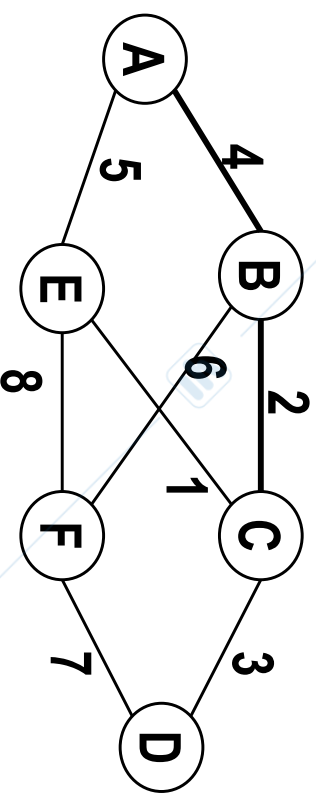
Pacchetti ricevuti

Network discovery

D	
C	3
F	7

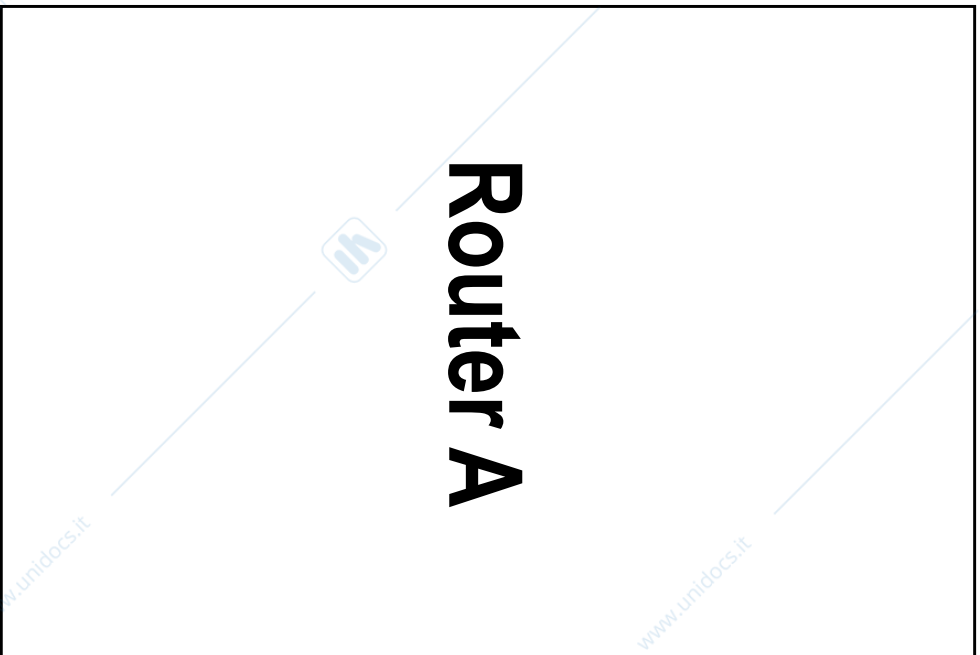


E	
A	5
C	1
F	8



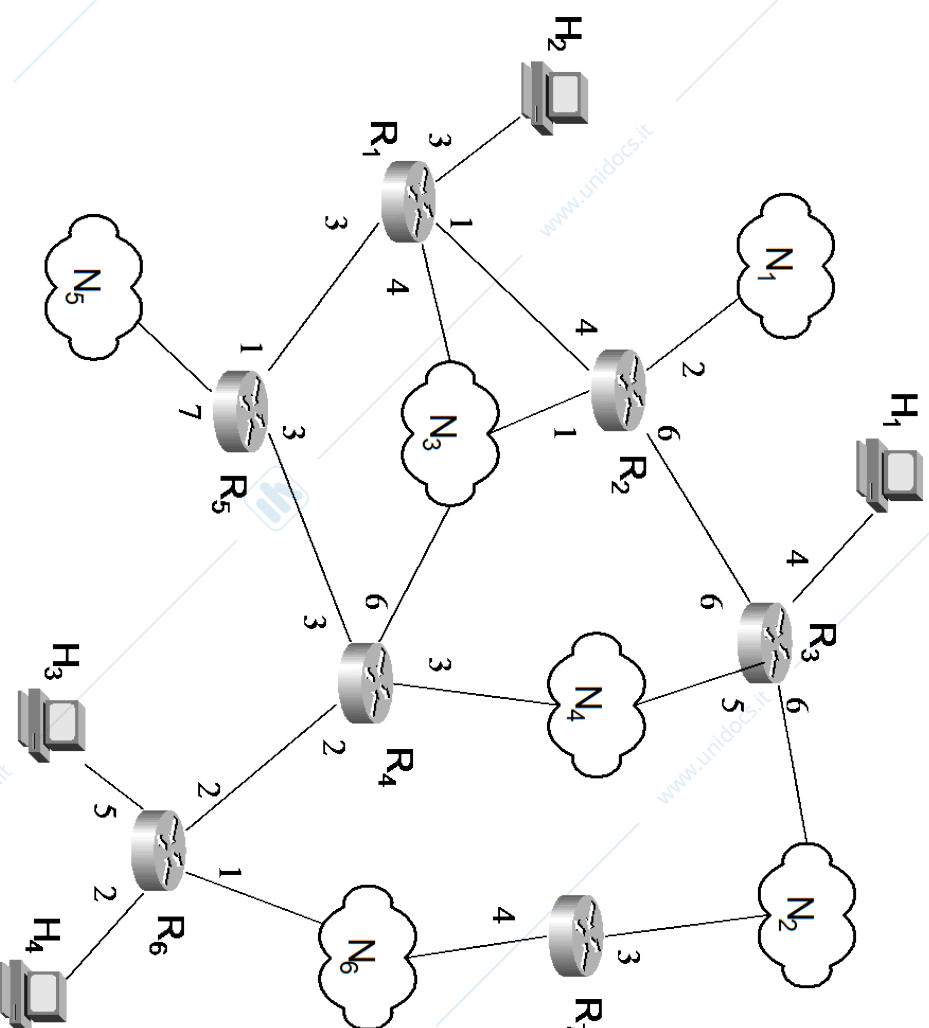
F	
B	6
D	7
E	8

Conferma



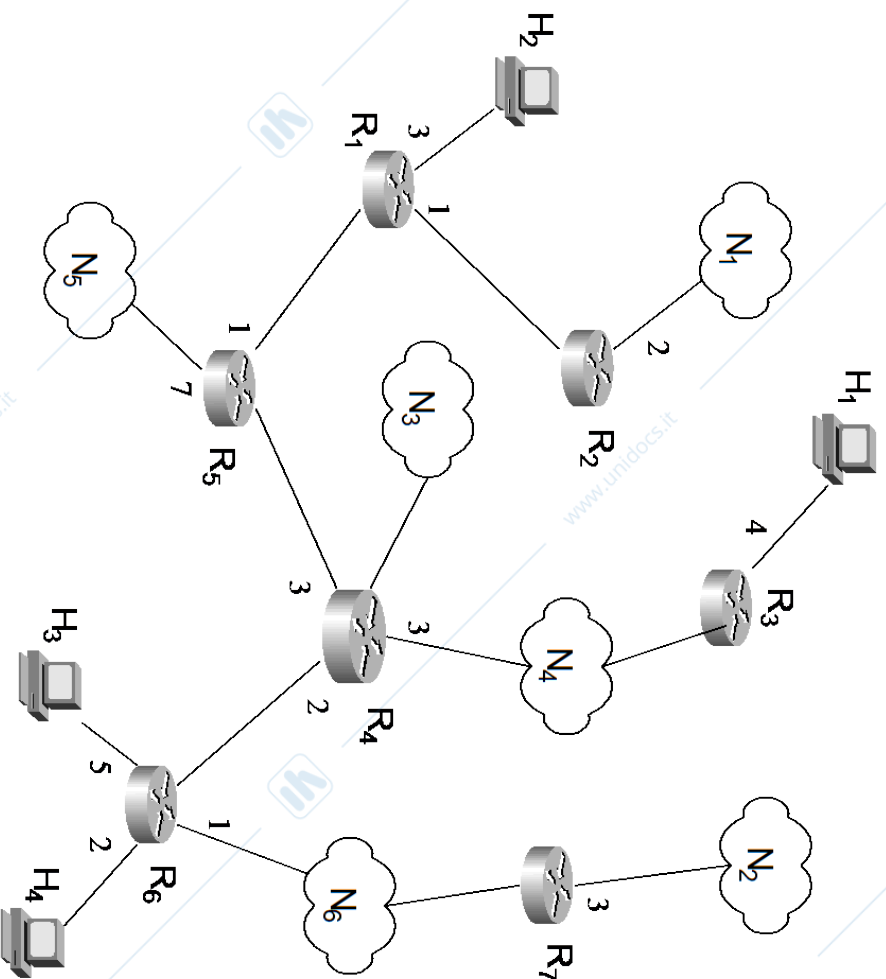
OSPF

Esempio



OSPF

Esempio



Destination.	Next hop	Cost.
N ₁	R ₅	7
N ₂	R ₆	6
N ₃	-	-
N ₄	-	-
N ₅	R ₅	10
N ₆	R ₆	3
H ₁	R ₃	7
H ₂	R ₅	7
H ₃	R ₆	7
H ₄	R ₆	4

- MST “visto” dal router R4 e tabella di routing corrispondente

■ FINE DEL MATERIALE PER L'ESAME