

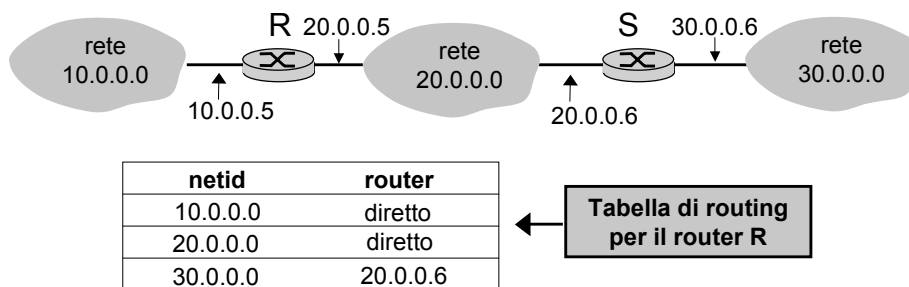
# Routing IP

## IP routing

- **IP routing (inoltro IP)**: meccanismo per la scelta del percorso in Internet attraverso il quale inviare i datagram
- IP routing effettuato dai **router** (scelgono il percorso)
- Routing diretto ed indiretto
  - Routing **diretto (direct delivery)**: nodo mittente e nodo destinatario appartenenti alla stessa rete fisica
  - Routing **indiretto (indirect delivery)**: il nodo mittente deve individuare un router a cui inviare il datagram
- Ogni router possiede una tabella di routing
  - **Tabella di routing**: memorizza le informazioni per raggiungere le possibili reti di destinazione (il router usa soltanto il **netid** dell'indirizzo IP del destinatario)

## IP routing (2)

- **Next-hop routing** (next-hop forwarding): il router possiede l'informazione sul salto successivo (*next hop*) che il datagram deve compiere per giungere a destinazione
- Il next-hop router appartiene ad una rete alla quale il router corrente è collegato direttamente



M. Colajanni, F. Mazzoni, 2003, Routing IP

5.2

## IP routing (3)

- Caratteristiche dell'IP routing:
  - **Indipendenza dal mittente**: il next-hop routing non dipende dal mittente del datagram o dal cammino che il datagram ha attraversato fino a quel momento
    - Il router estrae dal datagram soltanto l'indirizzo del destinatario
  - **Routing universale**: la tabella di routing deve contenere un next-hop router per ciascuna destinazione
  - **Routing ottimo**: il next-hop router deve essere scelto in modo da minimizzare il cammino verso la destinazione → **algoritmi di routing**
  - **Router di default**: per ridurre la dimensione della tabella di routing, si definisce un router di default comune a più indirizzi di destinazione

M. Colajanni, F. Mazzoni, 2003, Routing IP

5.3

## IP routing (4)

- Caratteristiche dell'IP routing (segue):
  - **Routing statico**: la tabella di routing non viene modificata dal router
    - l'amministratore di rete può inserire o modificare righe della tabella di routing
    - *svantaggio*: impossibilità di reagire automaticamente ai cambiamenti topologici
  - **Routing dinamico**: la tabella di routing viene modificata dal router al variare delle condizioni sulla rete (stato di funzionamento degli apparati e dei collegamenti)
    - lo scambio di informazioni ed il calcolo della tabella di routing avvengono tramite un algoritmo di routing

## IP routing (5)

- Funzionamento del router:
  1. Estrai l'indirizzo IP del destinatario D dal datagram e determina il suo netid N
  2. Se N corrisponde ad una rete connessa direttamente, consegna il datagram al destinatario D sulla rete (ciò comporta la risoluzione di D nel corrispondente indirizzo fisico e l'invio del frame)
  3. Altrimenti se la tabella contiene un router per la rete N, invia il datagram al next-hop router specificato nella tabella
  4. Altrimenti se la tabella contiene un router di default, invia il datagram a quel router
  5. Altrimenti dichiara un errore di routing

## Esempi di IP routing

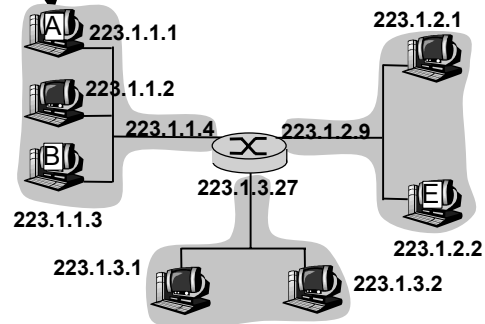
IP datagram:

Campi	IP addr sorgente	IP addr destinaz.	Dati
vari			

- Il datagram non viene modificato nel tragitto da mittente a destinatario
- campi addr di interesse

tabella di routing in A

Dest. net	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



M. Colajanni, F. Mazzoni, 2003, Routing IP

5.6

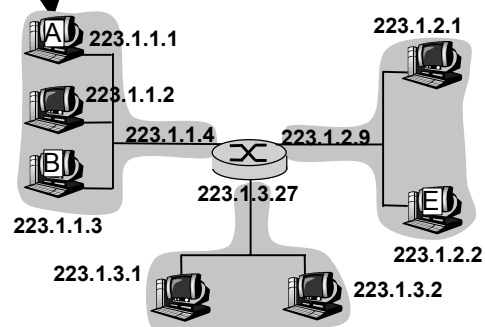
## Esempi di IP routing (2)

Campi	223.1.1.1	223.1.1.3	Dati
vari			

Partendo da A, se il datagram è destinato a B:

- ricerca l'indirizzo di rete di B
- B è sulla stessa rete di A
- il layer di collegamento dati invia il datagram direttamente a B incapsulandolo in un frame

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



M. Colajanni, F. Mazzoni, 2003, Routing IP

5.7

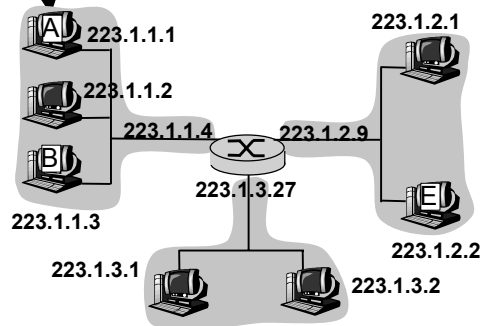
## Esempi di IP routing (3)

Campi vari	223.1.1.1	223.1.2.3	Dati
------------	-----------	-----------	------

Partendo da A, destinazione E:

- ricerca l'indirizzo di rete di E
- E è su una rete diversa
- routing table: next hop router per E è 223.1.1.4
- il layer di collegamento dati invia il datagram al router 223.1.1.4 nel frame
- il datagram arriva a 223.1.1.4
- *segue...*

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



M. Colajanni, F. Mazzoni, 2003, Routing IP

5.8

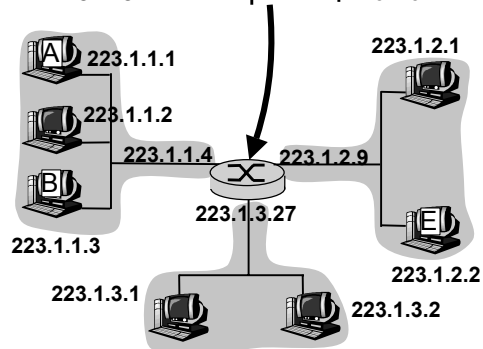
## Esempi di IP routing (4)

Campi vari	223.1.1.1	223.1.2.3	Dati
------------	-----------	-----------	------

Arrivato a 223.1.1.4, destinato a 223.1.2.2

- ricerca l'indirizzo di rete di E
- E sulla stessa rete dell'interfaccia di router 223.1.2.9
- il layer di collegamento dati invia il datagram a 223.1.2.2 incapsulandolo in un frame
- il datagram arriva a 223.1.2.2

Dest. Network	next router	Nhops	Interfaccia
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	-	1	223.1.3.27



M. Colajanni, F. Mazzoni, 2003, Routing IP

5.9

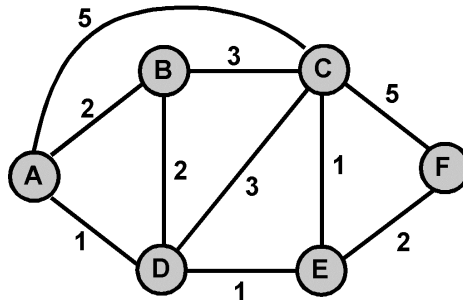
## Algoritmi di routing

• **Algoritmo di routing:** determinazione del percorso (path o route) ottimale (dato un insieme di router interconnessi, determinare il cammino ottimale dal mittente al destinatario)

- Cammino ottimale → costo minimo
- Per formulare un algoritmo di routing, si modella la rete tramite un grafo, i cui nodi rappresentano i router e gli archi le connessioni fisiche tra i router

Costo di inviare un pacchetto: etichetta sull'arco

Costo: livello di congestione del link, oppure distanza fisica, etc ...



M. Colajanni, F. Mazzoni, 2003, Routing IP

5.10

## Algoritmi di routing (2)

• Vincoli per la determinazione del cammino di costo minimo:

- il primo link è collegato direttamente al mittente
- l'ultimo link è collegato direttamente al destinatario
- $\forall i$ , l' $i$ -esimo e l' $i-1$ -esimo link sono connessi allo stesso nodo

• **cammino di costo minimo:** la somma dei costi dei link del cammino è la minima di tutti i possibili cammini dal mittente al destinatario

• **cammino minimo** (tutti gli archi hanno lo stesso peso): il cammino formato dal minimo numero di archi dal mittente al destinatario.

- Nel grafo nella figura precedente il cammino di costo minimo tra A e C è ADEC (ha costo pari a 3)

M. Colajanni, F. Mazzoni, 2003, Routing IP

5.11

## Algoritmi di routing (3)

- Classificazione degli algoritmi di routing:
  - **algoritmi di routing globale** (o centralizzati)
  - **algoritmi di routing distribuiti** (o decentralizzati)
- Algoritmi di routing globale:
  - Il cammino di costo minimo è calcolato avendo un'informazione globale sulla rete
  - Il calcolo può essere centralizzato su un unico nodo o replicato su più nodi
  - Noti come **link state algorithm**, dal momento che l'algoritmo deve conoscere lo stato (il costo) di ciascun link

M. Colajanni, F. Mazzoni, 2003, Routing IP

5.12

## Algoritmi di routing (4)

- Algoritmi di routing distribuiti:
  - Nessun nodo ha un'informazione completa del costo di tutti i link della rete
  - Il calcolo è distribuito ed interattivo
  - Ogni nodo inizia il calcolo conoscendo soltanto il costo dei link diretti: attraverso uno scambio di messaggi con i nodi vicini calcola il costo complessivo
  - Algoritmo del **vettore di distanza** (distance vector algorithm)
- Ulteriore classificazione degli algoritmi di routing:
  - algoritmi di routing **statici**
  - algoritmi di routing **dinamici**
- Algoritmi di routing usati in Internet:
  - algoritmo *dinamico* globale sullo stato dei link
  - algoritmo *dinamico* distribuito del vettore di distanza

M. Colajanni, F. Mazzoni, 2003, Routing IP

5.13

## Link state algorithm

- Topologia della rete, costi dei link conosciuti; cioè disponibili come input all'algoritmo
- Ciascun nodo trasmette le identità e i costi dei link ad esso collegati a tutti i router della rete.
- I nodi non devono conoscere inizialmente le identità di tutti gli altri nodi della rete.
- Alla fine i nodi conoscono tutti gli altri nodi ed i relativi percorsi con i loro costi
- L'algoritmo è conosciuto come algoritmo di Dijkstra ed è iterativo

## Link state algorithm (2) Notazioni

- $c(i,j)$  = costo del link dal nodo  $i$  al nodo  $j$   
Se  $i$  e  $j$  non sono collegati, si assume  $c(i,j) = \infty$   
Inoltre assumiamo  $c(i,j)=c(j,i)$
- $D(v)$  = costo del percorso dal nodo sorgente al nodo di destinazione  $v$  che ha attualmente (per questa iterazione) il minor costo
- $p(v)$  = il nodo precedente (collegato a  $v$ ) lungo il percorso attuale di minor costo dalla sorgente a  $v$
- $N$  = gruppo di nodi il cui percorso di minor costo è conosciuto definitivamente dalla sorgente



## Link state algorithm (3) Algoritmo

- Inizializzazione:**

```

N = {A}
per tutti i nodi v
  se v è adiacente ad A
    allora D(v) = c(A,v)
  altrimenti D(v) = ∞
  
```

- Ciclo:**

```

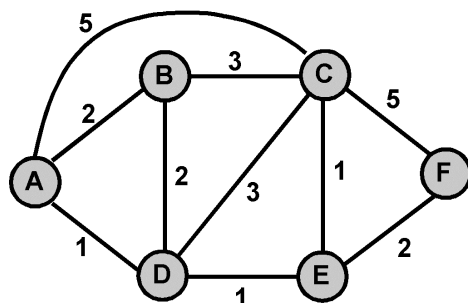
trova un nodo w ∉ N tale che D(w) è minimo
aggiungi w ad N
aggiorna D(v) per tutti i v adiacenti a w e v ∉ N
D(v) = min(D(v), D(w) + c(w,v))
/*il nuovo costo per v è o il vecchio costo per v o
il costo del minor cammino conosciuto per w più il
costo da w a v*/
finché tutti i nodi ∈ N
  
```

M. Colajanni, F. Mazzoni, 2003, Routing IP

5.16

## Link state algorithm (4) Esempio

- Calcoliamo i percorsi di minor costo da A a tutte le possibili destinazioni.
- Nel passo di inizializzazione si costruisce la seguente tabella:



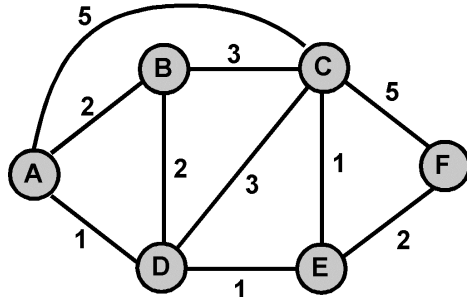
Passo	N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	∞	∞

M. Colajanni, F. Mazzoni, 2003, Routing IP

5.17

## Link state algorithm (5) Esempio (segue)

- Nei passi successivi si riempie la tabella, secondo le regole esposte in precedenza e si ottiene:



Passo	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	$\infty$	$\infty$
1	AD	2,A	4,D	1,A	2,D	$\infty$
2	ADE	2,A	3,E	1,A	2,D	4,E
3,4,5	ADEB(C(F))	2,A	3,E	1,A	2,D	4,E

M. Colajanni, F. Mazzoni, 2003, Routing IP

5.18

## Link state algorithm (6) Determinazione del percorso

- Quando l'algoritmo LS termina, abbiamo per ciascun nodo, il suo predecessore lungo il percorso di minor costo dalla sorgente.
- Per ciascun predecessore abbiamo il suo predecessore e così via, in modo da poter ricostruire l'intero percorso dalla sorgente a tutte le destinazioni.
- Ad esempio, se da A voglio arrivare ad F, guardo nella tabella e vedo che il costo è 4 e il predecessore di F è E.
- A questo punto cerco il predecessore di E che è D, poi il predecessore di D, che è A. Ho determinato che il percorso di costo minimo è ADEF.

M. Colajanni, F. Mazzoni, 2003, Routing IP

5.19

## Distance vector algorithm

- Il router tiene in una tabella tutti i percorsi di instradamento noti
- All'avvio, il router inizializza la sua tabella di routing per contenere una voce per ciascuna rete con cui è connesso direttamente.
- Ogni voce identifica una rete di destinazione e la relativa distanza, di solito misurata in salti o hop. Ogni hop si riferisce al numero di router che un datagram attraversa dalla propria sorgente alla destinazione

M. Colajanni, F. Mazzoni, 2003, Routing IP

5.20

## Distance vector algorithm (2)

- Periodicamente ogni router invia una copia della propria tabella di routing agli altri router che può raggiungere direttamente
- Quando dal router J arriva un resoconto al router K, questo esamina l'insieme di destinazioni riportate e le relative distanze
- Se J conosce un percorso più breve per raggiungere una destinazione, o se elenca una destinazione che K non ha nella sua tabella, o se K al momento sta inviando ad una destinazione tramite J e la distanza di J a quella destinazione è cambiata, K sostituisce la voce nella sua tabella.

M. Colajanni, F. Mazzoni, 2003, Routing IP

5.21

## Distance vector algorithm (3)

- **Notare che se J riporta la distanza N, una voce aggiornata in K avrà la distanza  $N+1$  (la distanza per raggiungere la destinazione da J più la distanza per raggiungere J)**
- **La tabella di routing contiene una terza colonna che specifica il salto successivo**
- **Quando il router K aggiunge o modifica una voce in risposta ad un messaggio proveniente da J, assegna il router J come il salto successivo per quella voce.**