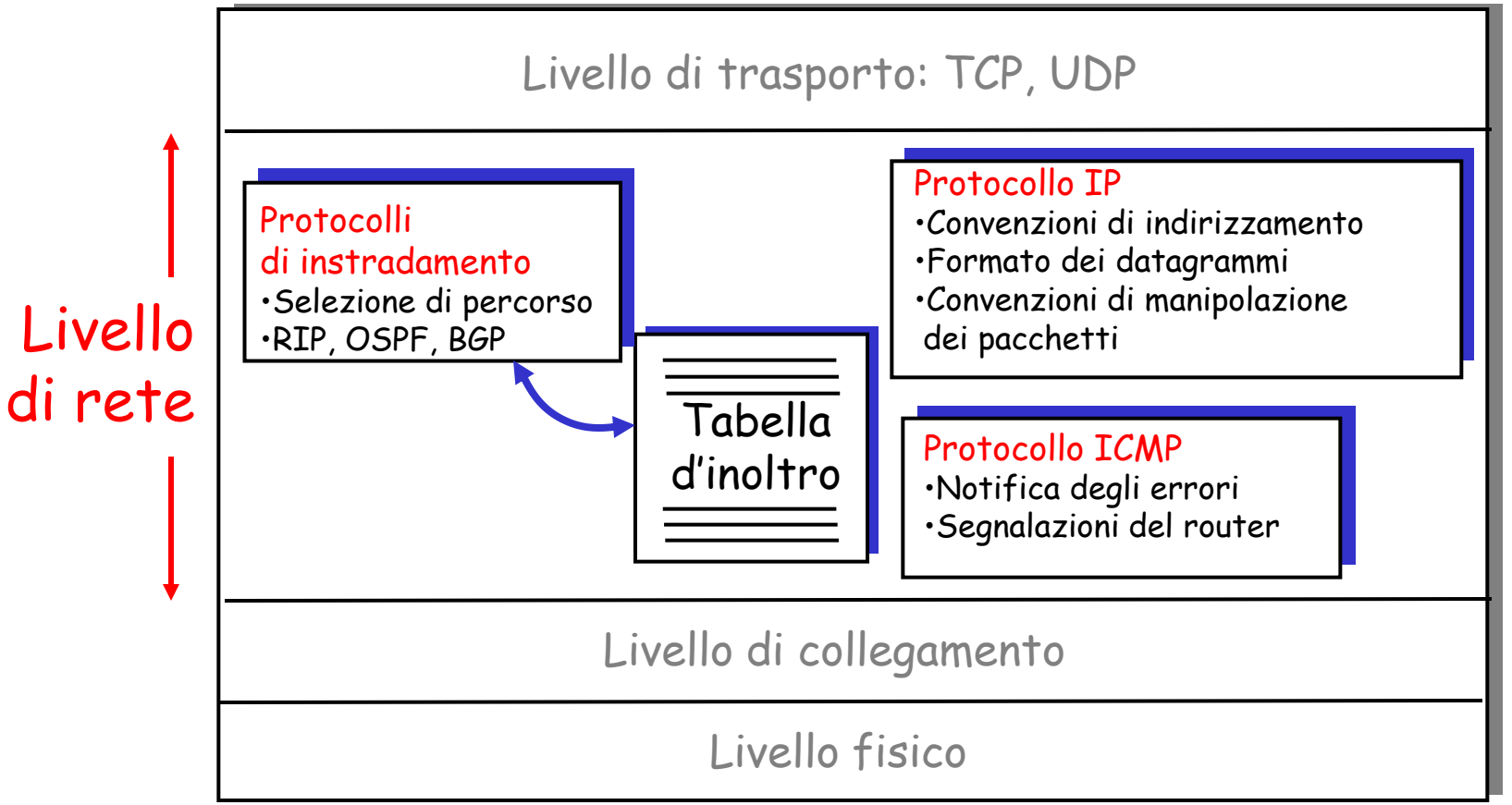
Livello di Rete: IPv6, Algoritmi di instradamento

Gaia Maselli
maselli@di.uniroma1.it

Livello di rete



Livello di rete

Introduzione

Reti a circuito virtuale e a datagramma

Che cosa si trova all'interno di un router?

Protocollo Internet (IP)

- Formato dei datagrammi
- Indirizzamento IPv4
- ICMP
- IPv6

Algoritmi di instradamento

- Stato del collegamento
- Vettore distanza
- Instradamento gerarchico

Instradamento in Internet

- RIP
- OSPF
- BGP

Instradamento broadcast e multicast

IPv6

- Esigenza principale: lo spazio di indirizzamento IP a 32 bit stava incominciando a esaurirsi (in IPv6 indirizzi a 128 bit).
- Altre motivazioni:
  - Il formato dell'intestazione aiuta a rendere più veloci i processi di elaborazione e inoltro
  - Agevolare la QoS.

Formato dei datagrammi IPv6

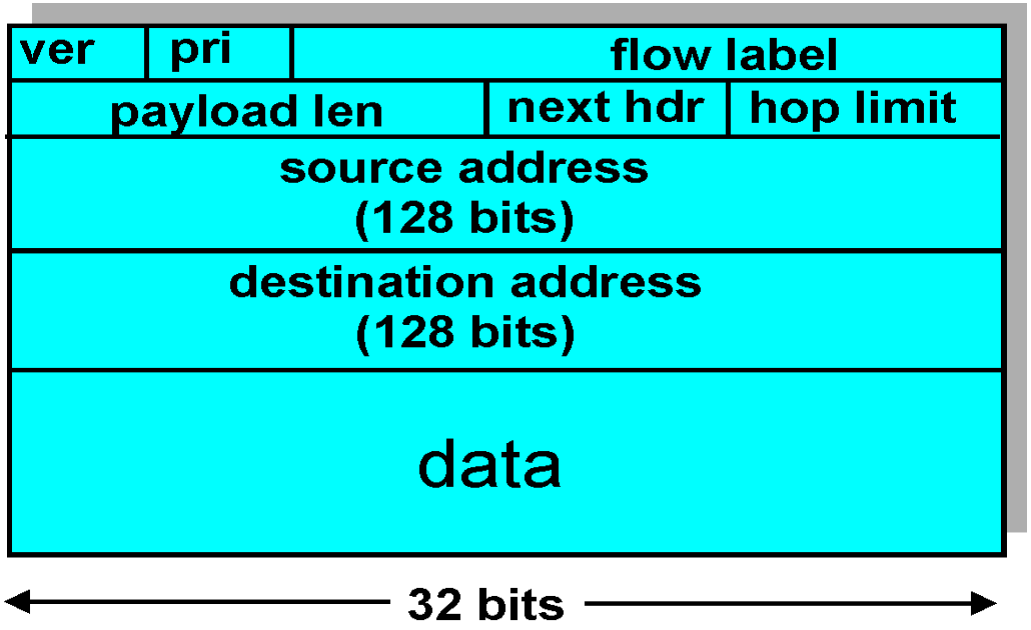
Formato dei datagrammi IPv6:

- Intestazione a 40 byte e a lunghezza fissa.

Priorità di flusso: attribuisce priorità a determinati datagrammi di un flusso.

Etichetta di flusso: identifica i pacchetti che appartengono a flussi particolari (anche se non è ben chiaro il concetto di "flusso").

Intestazione successiva: identifica il protocollo cui verranno consegnati i contenuti del datagramma.



Altre novità di IPv6

- *Checksum*: i progettisti hanno deciso di rimuoverla dal livello di rete in quanto risultava ridondante.
  - Richiede di essere aggiornata a ogni hop a causa del campo TTL in IPv4
- Non è consentita la frammentazione.
- *Opzioni:* non fa più parte dell'intestazione IP standard. Il campo non è del tutto scomparso ma è diventato una delle possibili "intestazioni successive" cui punta l'intestazione di IPv6.
- *ICMPv6:* nuova versione di ICMP:
  - Ha aggiunto nuovi tipi e codici, es. "Pacchetto troppo grande".
  - Assume le funzionalità dell'IGMP, e gestisce l'ingresso e l'uscita di host nei gruppi multicast.

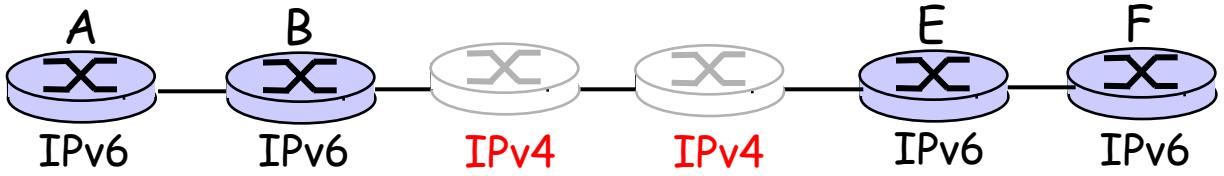
Passaggio da IPv4 a IPv6

- Non è possibile aggiornare simultaneamente tutti i router:
  - Impossibile dichiarare una "giornata campale" in cui tutte le macchine Internet verranno spente e aggiornate da IPv4 a IPv6.
  - Come riuscirà la rete a funzionare in presenza di router IPv4 e IPv6?
- *Tunneling:* IPv6 viene trasportato come payload in datagrammi IPv4 quando attraversa router IPv4

Tunneling

Vista logica:

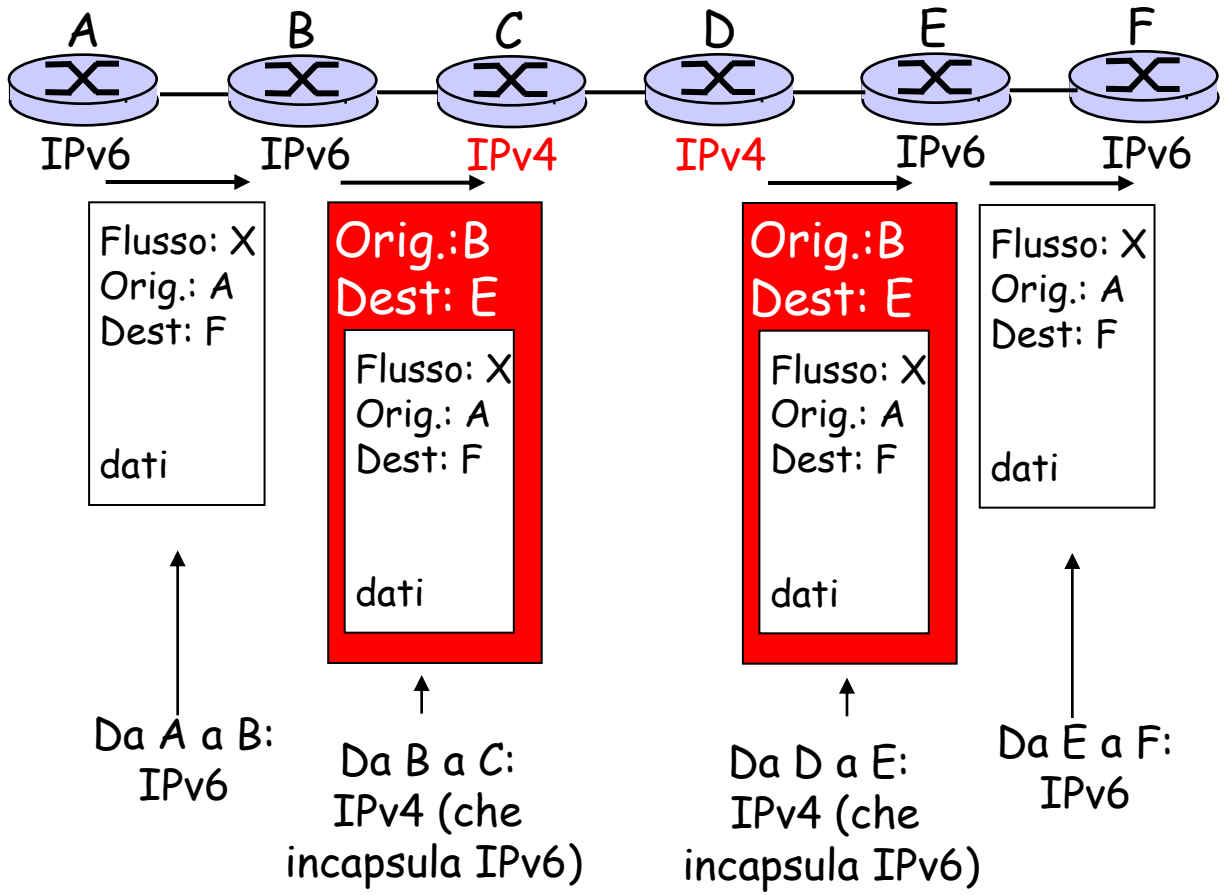
Vista fisica:



Tunneling

Vista logica:

Vista fisica:



Livello di rete

Introduzione

Reti a circuito virtuale e a datagramma

Che cosa si trova all'interno di un router?

Protocollo Internet (IP)

- Formato dei datagrammi
- Indirizzamento IPv4
- ICMP
- IPv6

Algoritmi di instradamento

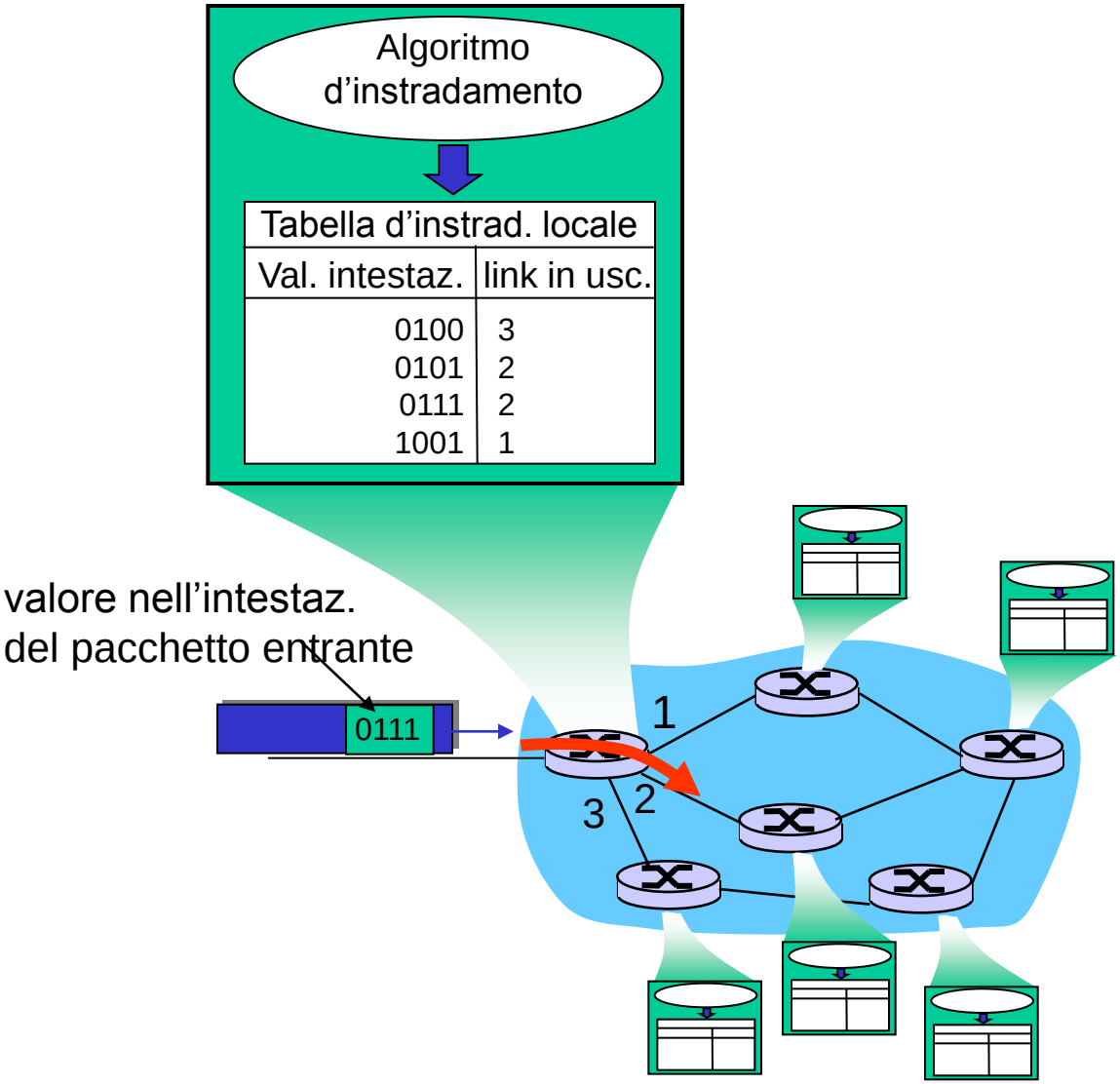
- Stato del collegamento
- Vettore distanza
- Instradamento gerarchico

Instradamento in Internet

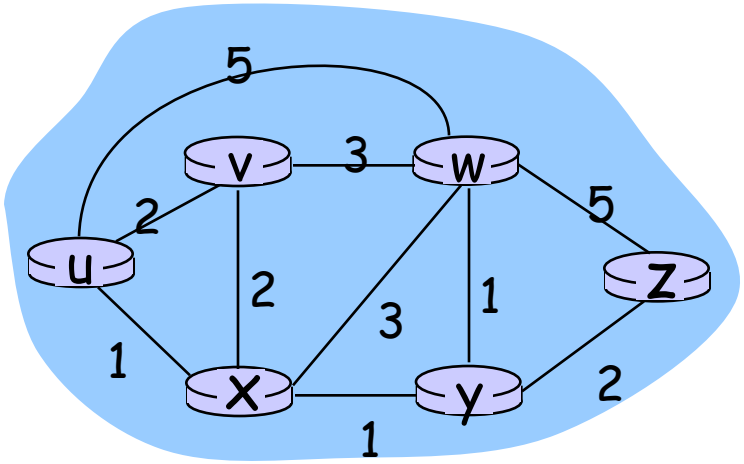
- RIP
- OSPF
- BGP

Instradamento broadcast e multicast

Algoritmi d'instradamento (routing)



Grafo di una rete di calcolatori



Grafo: G = (N,E)

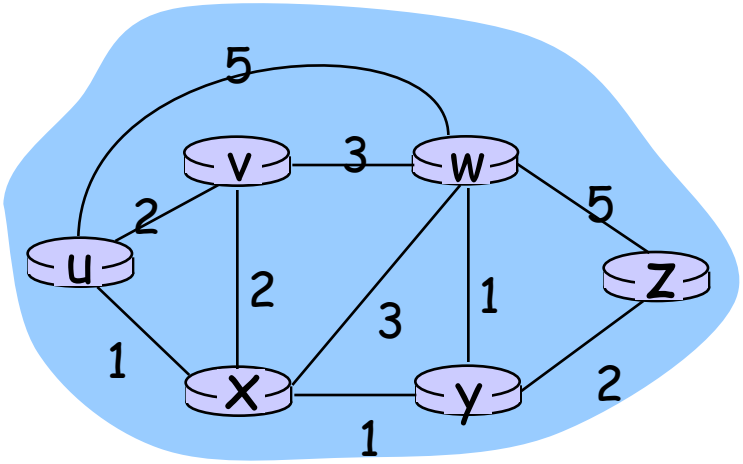
N = insieme di nodi (router) = { u, v, w, x, y, z }

E = insieme di archi (collegamenti) ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

N.B.: Il grafo è un'astrazione utile anche in altri contesti di rete

Esempio: P2P, dove N è un insieme di peer ed E è un insieme di collegamenti TCP

Grafo di una rete : costi



- c(x,x') = costo del collegamento (x,x')

  - es., c(w,z) = 5

- il costo di un cammino è semplicemente la somma di tutti i costi degli archi lungo il cammino

Costo di un cammino $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Domanda: Qual è il cammino a costo minimo tra u e z ?

Algoritmo d'instradamento: determina il cammino a costo minimo.

Classificazione degli algoritmi d'instradamento

Globale o decentralizzato?

Globale:

- L'algoritmo riceve in ingresso tutti i collegamenti tra i nodi e i loro costi.
- Algoritmi a stato del collegamento (*link-state algorithm).*

Decentralizzato:

- Ogni nodo elabora un vettore di stima dei costi (distanze) verso tutti gli altri nodi nella rete.
- Il cammino a costo minimo viene calcolato in modo distribuito e iterativo.
- Algoritmo a vettore distanza (VC, *distance-vector algorithms*)

Statico o dinamico?

Statico:

- I cammini cambiano molto raramente.

Dinamico:

- Determinano gli instradamenti al variare di:
  - Volume di traffico
  - Topologia della rete

Livello di rete

Introduzione

Reti a circuito virtuale e a datagramma

Che cosa si trova all'interno di un router?

Protocollo Internet (IP)

- Formato dei datagrammi
- Indirizzamento IPv4
- ICMP
- IPv6

Algoritmi di instradamento

- Stato del collegamento
- Vettore distanza
- Instradamento gerarchico

Instradamento in Internet

- RIP
- OSPF
- BGP

Instradamento broadcast e multicast

Algoritmo d'instradamento a stato del collegamento (LS)

Algoritmo di Dijkstra:

- La topologia di rete e tutti i costi dei collegamenti sono noti a tutti i nodi
  - attraverso il "link-state broadcast".
  - tutti i nodi dispongono delle stesse informazioni
- Calcola il cammino a costo minimo da un nodo (origine) a tutti gli altri nodi della rete.
  - Crea una tabella d'inoltro per quel nodo
- È iterativo: dopo la *k*-esima iterazione i cammini a costo minimo sono noti a *k* nodi di destinazione.

Definiamo la seguente notazione:

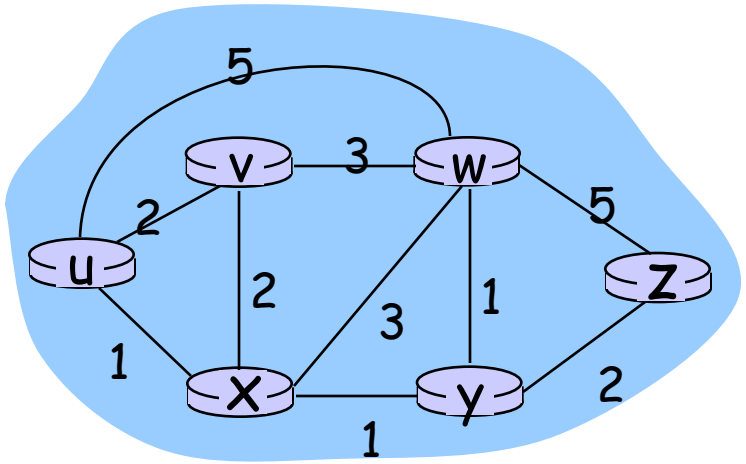
- c(x,y): costo del collegamenti dal nodo x al nodo y
- D(v): costo del cammino dal nodo origine alla destinazione *v* per quanto riguarda l'iterazione corrente.
- p(v): immediato predecessore di *v* lungo il cammino.
- N': sottoinsieme di nodi per cui il cammino a costo minimo dall'origine è definitivamente noto.

Algoritmo di Dijsktra

```
1  Inizializzazione:
2    N' = {u}
3    per tutti i nodi v
4      se v è adiacente a u
5          allora D(v) = c(u,v)
6      altrimenti D(v) = ∞
7
8  Ciclo
9      determina un w non in N' tale che D(w) sia minimo
10     aggiungi w a N'
11     aggiorna D(v) per ciascun nodo v adiacente a w e non in N' :
12        D(v) = min( D(v), D(w) + c(w,v) )
13     /* il nuovo costo verso v è il vecchio costo verso v oppure
14      il costo del cammino minimo noto verso w più il costo da w a v */
15 Finché N' = N
```

Algoritmo di Dijkstra: esempio

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					



Algoritmo di Dijkstra: un altro esempio

Cammino a costo minimo da u:

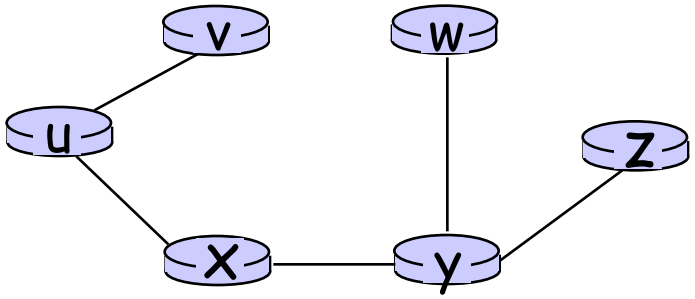


Tabella d'inoltro in u:

destinazione	collegamento
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Livello di rete

- Introduzione
- Reti a circuito virtuale e a datagramma
- Che cosa si trova all'interno di un router?
- Protocollo Internet (IP)
  - Formato dei datagrammi
  - Indirizzamento IPv4
  - ICMP
  - IPv6
- Algoritmi di instradamento
  - Stato del collegamento
  - Vettore distanza
  - Instradamento gerarchico
- Instradamento in Internet
  - RIP
  - OSPF
  - BGP
- Instradamento broadcast e multicast

Algoritmo d'instradamento con vettore distanza (DV)

Distribuito: ogni nodo riceve informazione dai vicini e opera su quelle

Asincrono: non richiede che tutti i nodi operino al passo con gli altri

Formula di Bellman-Ford

definisce

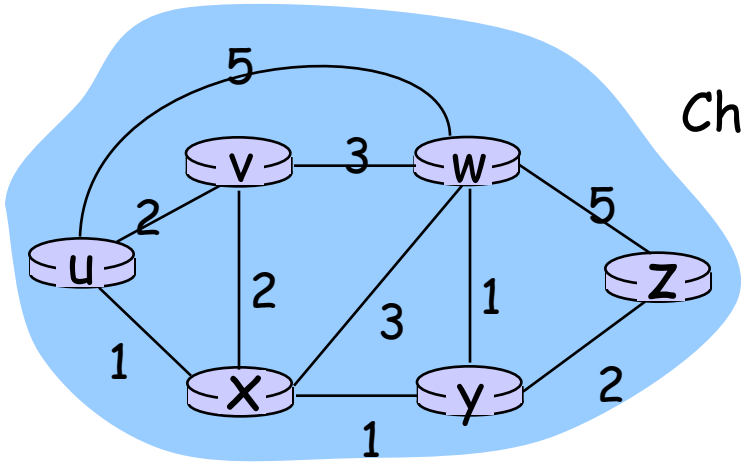
$d_x(y)$:= il costo del percorso a costo minimo dal nodo *x* al nodo *y*.

Allora:

$$d_x(y) = \min_v\{c(x,v) + d_v(y)\}$$

dove min_v riguarda tutti i vicini di *x*.

Formula di Bellman-Ford: esempio



Chiaramente, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

La formula B-F definisce:

$$d_u(z) = \min \{ c(u,v) + d_v(z), \; c(u,x) + d_x(z), \; c(u,w) + d_w(z) \}$$
$$= \min \{2 + 5, \; 1 + 3, \; 5 + 3\} = 4$$

Algoritmo con vettore distanza

- Consideriamo il nodo x
- $D_x(y)$ = stima del costo del percorso a costo minimo da se stesso al nodo *y*.
- Vettore distanza: $\mathbf{D}_x = [D_x(y): y \text{ in } N]$

  stima presso x del costo verso tutte le destinazioni y in N

- Il nodo x conosce il costo verso ciascun vicino *v*: c(x,v)
- Il nodo *x* mantiene $\mathbf{D}_x = [D_x(y): y \text{ in } N]$
- Il nodo x mantiene anche i vettori distanza di ciascuno dei suoi vicini
  - Per ciascun vicino v, x mantiene $\mathbf{D}_v = [D_v(y): y \text{ in } N]$

Algoritmo con vettore distanza

Idea di base:

- Ogni nodo invia una copia del proprio vettore distanza a ciascuno dei suoi vicini.
- Quando un nodo x riceve un nuovo vettore distanza, DV, da qualcuno dei sui vicini, lo salva e usa la formula B-F per aggiornare in proprio vettore distanza come segue:

$$D_x(y) \leftarrow min_v\{c(x,v) + D_v(y)\} \quad \text{per ciascun nodo } y \text{ in } N.$$

- Se il vettore distanza del nodo *x* è cambiato per via di tale passo di aggiornamento, il nodo *x* manderà il proprio vettore distanza aggiornato a ciascuno dei suoi vicini, i quali a loro volta aggiornano il loro vettore distanza

Algoritmo con vettore distanza

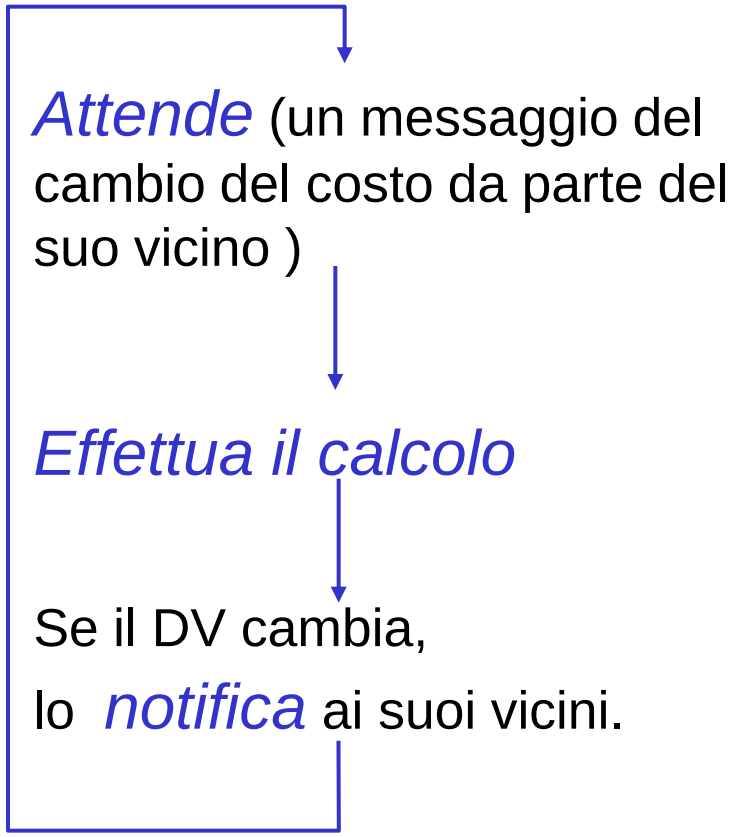
Iterativo, asincrono: ogni iterazione locale è causata da:

- cambio del costo di uno dei collegamenti locali
- ricezione da qualche vicino di un vettore distanza aggiornato.

Distribuito:

- Ogni nodo aggiorna i suoi vicini *solo* quando il suo DV cambia.
  - i vicini avvisano i vicini solo se necessario.

Ciascun nodo:



Algoritmo Distance Vector

A ciascun nodo x:

Inizializzazione

```
per tutte le destinazioni y in N:
    Dx(y) = c(x,y)        //c(x,y)=∞ se y non vicino
per ciascun vicino w
    Dw(y) = ∞ per tutte le destinazioni y in N
per ciascun vicino w
    invia il vettore distanza Dx = [Dx(y): y in N] a w
```

Ciclo

```
attendi (finchè vedi cambiare il costo di un collegamento
    verso qualche vicino w o ricevi un vettore distanza
    da qualche vicino w)
per ogni y in N:
    Dx(y) = minv {c(x,y) + Dv(y)}
se qualche Dx(y) è cambiato invia vettore distanza Dx a
tutti i vicini
```

$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$
$= \min\{2+0\ ,\ 7+1\} = 2$

$D_x(z) = \min\{c(x,y)+D_y(z),\ c(x,z) + D_z(z)\}$
$= \min\{2+1\ ,\ 7+0\} = 3$

Tabella del nodo x

da \ costo verso	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

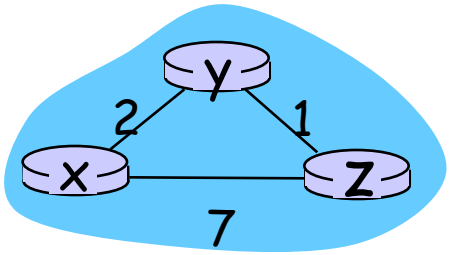
da \ costo verso	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

Tabella del nodo y

da \ costo verso	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

Tabella del nodo z

da \ costo verso	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0



tempo

$$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$$
$$= \min\{2+0\ ,\ 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\}$$
$$= \min\{2+1\ ,\ 7+0\} = 3$$

Tabella del nodo x

da \ costo verso	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

da \ costo verso	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

da \ costo verso	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

Tabella del nodo y

da \ costo verso	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

da \ costo verso	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

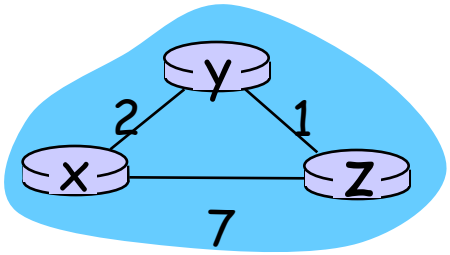
da \ costo verso	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

Tabella del nodo z

da \ costo verso	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

da \ costo verso	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

da \ costo verso	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

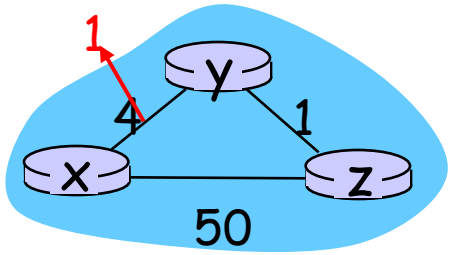


tempo

Algoritmo con vettore distanza: modifica dei costi

Modifica dei costi:

- Un nodo rileva un cambiamento nel costo dei collegamenti.
- Aggiorna il proprio vettore distanza.
- Se si verifica un cambiamento nel costo, trasmette ai suoi vicini il nuovo DV.



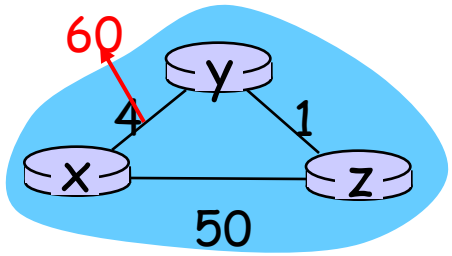
le buone notizie viaggiano in fretta

All'istante t_0, y rileva il cambiamento nel costo del collegamento, aggiorna il proprio DV e informa i vicini del cambiamento.

All'istante t_1, z riceve l'aggiornamento da y e aggiorna la propria tabella, calcola un nuovo costo minimo verso x e invia il nuovo DV ai vicini.

All'istante t_2, y riceve l'aggiornamento di z e aggiorna la propria tabella di distanza. I costi minimi di y non cambiano e y non manda alcun messaggio a z.

Algoritmo con vettore distanza: modifica dei costi



Modifica dei costi:

- Le buone notizie (costo diminuito) si sono propagate rapidamente.
- Le cattive notizie si propagano lentamente: problema dell'instradamento ciclico!
- 44 iterazioni prima che l'algoritmo di stabilizzi

  (provare l'algoritmo per esercizio)

Inversione avvelenata:

- Se Z instrada tramite Y per giungere alla destinazione X :
  - Allora Z avvertirà Y che la sua distanza verso X è infinita (così Y non tenterà mai d'instradare verso X passando per Z)
- L'inversione avvelenata può risolvere il problema dei cicli?

Confronto tra gli algoritmi LS e DV

Complessità dei messaggi:

- LS: con n nodi, E collegamenti, implica l'invio di $O(nE)$ messaggi.
- DV: richiede scambi tra nodi adiacenti.
  - Il tempo di convergenza può variare.

Velocità di convergenza:

- LS: l'algoritmo $O(n^2)$ richiede $O(nE)$ messaggi.
  - ci possono essere oscillazioni di velocità.
- DV: può convergere lentamente.
  - può presentare cicli d'instradamento.
  - può presentare il problema del conteggio all'infinito.

Robustezza: cosa avviene se un router funziona male?

LS:

- un router può comunicare via broadcast un costo sbagliato per uno dei suoi collegamenti connessi (ma non per altri).
- i nodi si occupano di calcolare soltanto le proprie tabelle.

DV:

- un nodo può comunicare cammini a costo minimo errati a tutte le destinazioni.
- la tabella di ciascun nodo può essere usata degli altri.
  - ❿ Un calcolo errato si può diffondere per l'intera rete.