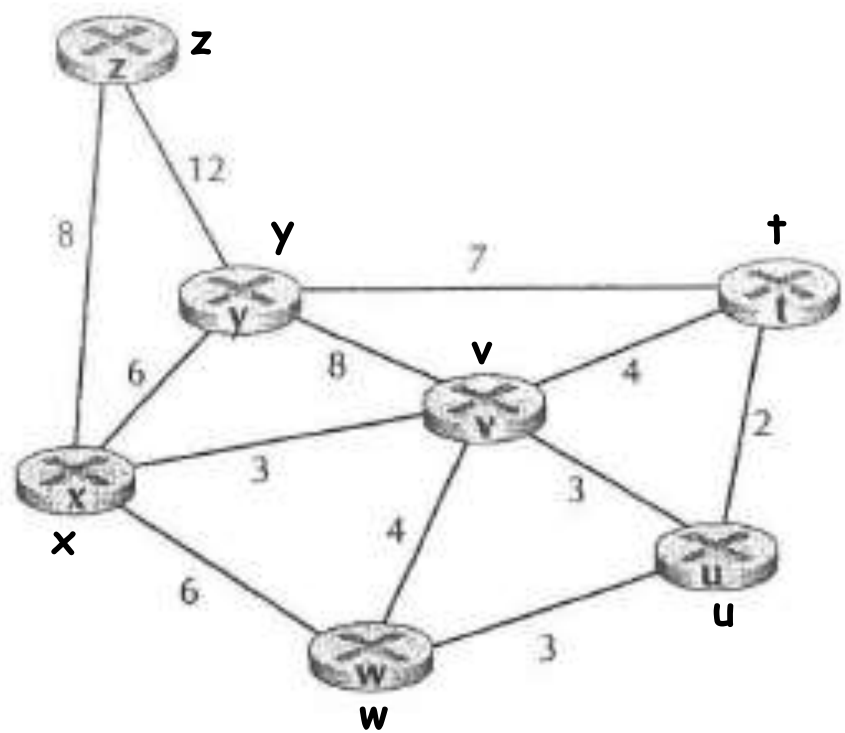
Marco Listanti

Esercizi strato di rete (parte 3)

Esercizio 1 (1)

- Si consideri la rete in figura
- Si utilizzi l'algoritmo di Dijkstra per calcolare il percorso più breve dal nodo **x** a tutti gli altri nodi della rete
- Si mostri il funzionamento dell'algoritmo e si determini la tabella di routing associata al nodo **x**



Esercizio 1 (2)

PASSO	INSIEME T_k	NODI					
		t	u	v	w	y	z
		D(t), p(t)	D(u), p(u)	D(v), p(v)	D(w), p(w)	D(y), p(y)	D(z), p(z)
0	x	∞	∞	3, x	6, x	6, x	8, x
1	xv	7, v	6, v		6, x	6, x	8, x
2	xvu	7, v			6, x	6, x	8, x
3	xuvw	7, v				6, x	8, x
4	xuvwy	7, v					8, x
5	xuvwyt						8, x
6	xuvwytz						

D(i) = costo minimo per percorso tra il nodo di origine ed il nodo "i"

p(i) = predecessore del nodo "i" sul percorso minimo tra in nodo origine il nodo "i"

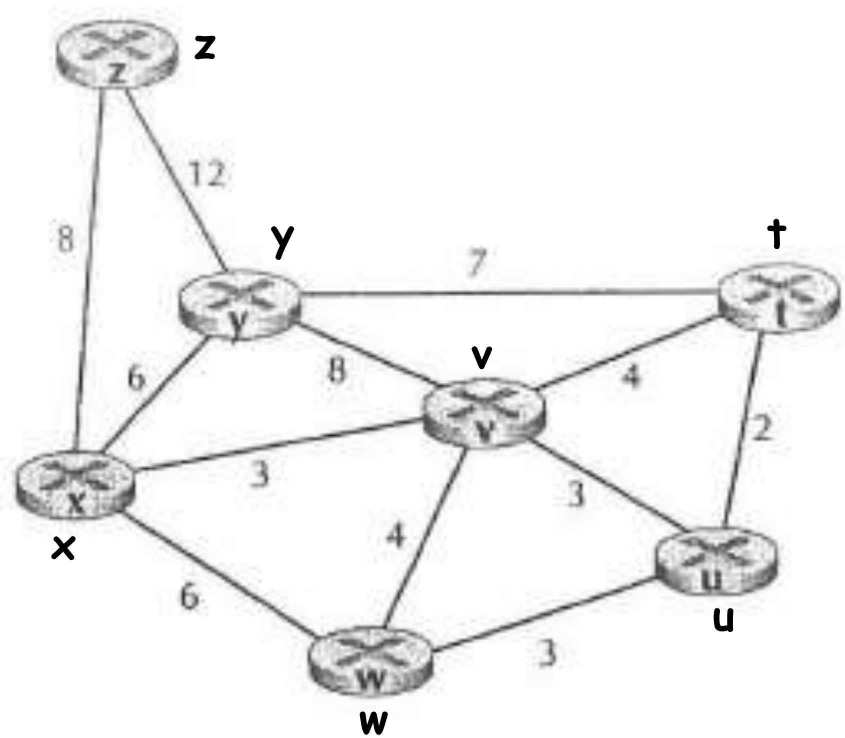
Esercizio 1 (3)

- La tabella di routing del nodo x sarà

Destinazione	Next Hop
Nodo **t**	V
Nodo **u**	V
Nodo **v**	V (locale)
Nodo **w**	W (locale)
Nodo **y**	Y (locale)
Noco **z**	Z (locale)

Esercizio 2 (1)

- Si considerino le stesse ipotesi dell'esercizio precedente
- Si determini lo spanning tree a costo minimo che ha come radice il nodo v
- Si determini inoltre la tabella di routing associata al nodo v



Esercizio 2 (2)

PASSO	T_k	NODI					
		t	u	x	w	y	z
		D(t), p(t)	D(u), p(u)	D(x), p(x)	D(w), p(w)	D(y), p(y)	D(z), p(z)
0	v	4, t	3, u	3, x	4, w	8, y	∞
1	vu	4, t		3, x	4, w	8, y	∞
2	vux	4, t			4, w	8, y	11, x
3	vuxt				4, w	8, y	11, x
4	vuxtw					8, y	11, x
5	vuxtwy						11, x
6	vuxtwyz						

D(i) = costo minimo per percorso tra il nodo di origine ed il nodo "i";

p(i) = predecessore del nodo "i" sul percorso minimo tra in nodo origine il nodo "i"

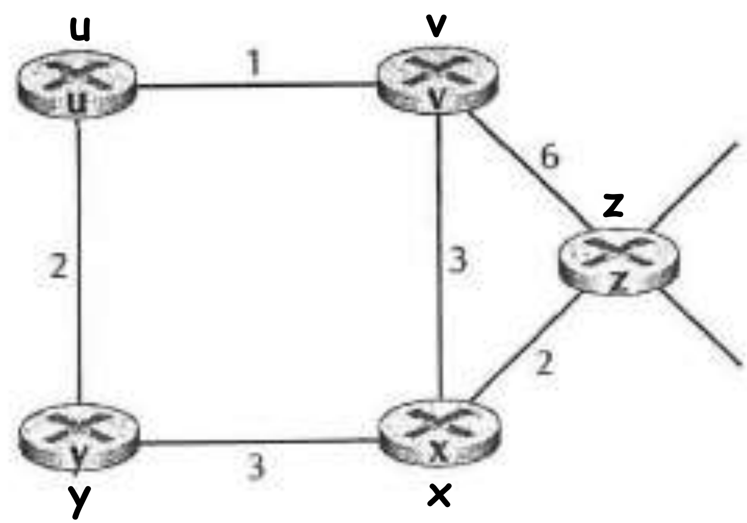
Esercizio 2 (3)

- La tabella di routing del nodo v sarà

Destinazione	Next Hop
Nodo **t**	T (locale)
Nodo **u**	U (locale)
Nodo **v**	V (locale)
Nodo **w**	W (locale)
Nodo **y**	Y (locale)
Noco **z**	Y

Esercizio 3 (1)

- Si consideri la rete in figura e si assuma che inizialmente ciascun nodo conosca solo il costo associato ai suoi archi incidenti
- Si consideri l'algoritmo "distance vector"
- Si determinino, passo dopo passo, le tabelle delle distanze dei nodi U, V, e Z



Esercizio 3 (2)

Condizioni iniziali

Cond. Iniziali		Nodi destinazione				
Nodi origine		U	V	X	Y	Z
	V	∞	0	∞	∞	∞
	X	∞	∞	0	∞	∞
	Z	∞	6	2	∞	0

Riceve il distance vector da X e V → Z

Passo 1		Nodi destinazione				
Nodi origine		U	V	X	Y	Z
	V	1	0	3	∞	6
	X	4	3	0	3	2
	Z	7	5	2	5	0

Riceve il distance vector da U, X e Z → V

Riceve il distance vector da Y, V e Z → X

Riceve il distance vector da X, V → Z

Esercizio 3 (3)

Passo 2		Nodi destinazione				
		U	V	X	Y	Z
Nodi origine	V	1	0	3	3	5
	X	4	3	0	3	2
	Z	6	5	2	5	0

Riceve il distance vector da X e V

Passo 3		Nodi destinazione				
		U	V	X	Y	Z
Nodi origine	V	1	0	3	3	5
	X	4	3	0	3	2
	Z	6	5	2	5	0

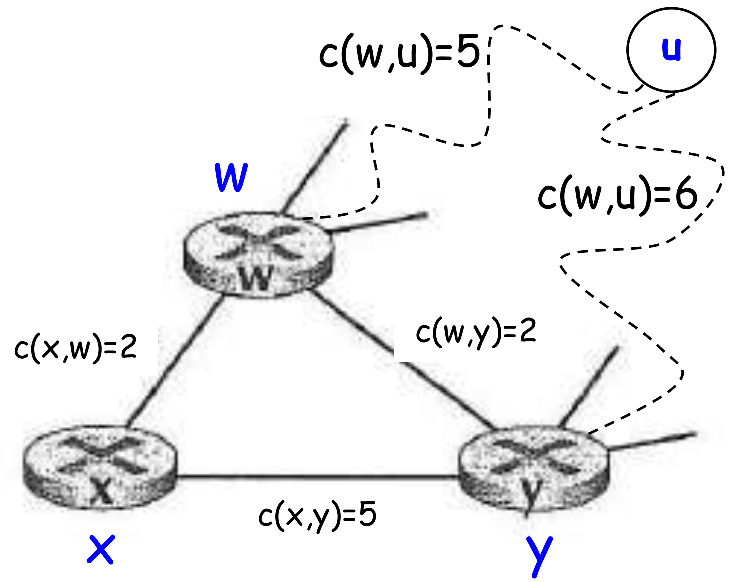
Riceve il distance vector da U, X e Z

Riceve il distance vector da Y, V e Z

Riceve il distance vector da X, V

Esercizio 4 (1)

- Si consideri il segmento di rete in figura
- Il nodo **x** è collegato solamente a **w** e **y**
  - Il nodo **w** ha un percorso minimale non noto verso la destinazione **u** di costo 5
  - Il nodo **y** ha un percorso minimale non noto verso **u** di costo 6
- (a) Si determini il vettore delle distanze di **x** per le destinazioni **w**, **y** e **u**
- (b) Si determinino le variazioni nel costo dei collegamenti sia per **c(x,w)** che per **c(x,y)** in modo tale che **x** informi i suoi vicini di un nuovo percorso a costo minimo verso **u**
- (c) Si determini una variazione nel costo dei collegamenti sia per **c(x,w)** che per **c(x,y)** tale che **x** non informi i suoi vicini di un nuovo percorso a costo minimo verso **u**



Esercizio 4 (2)

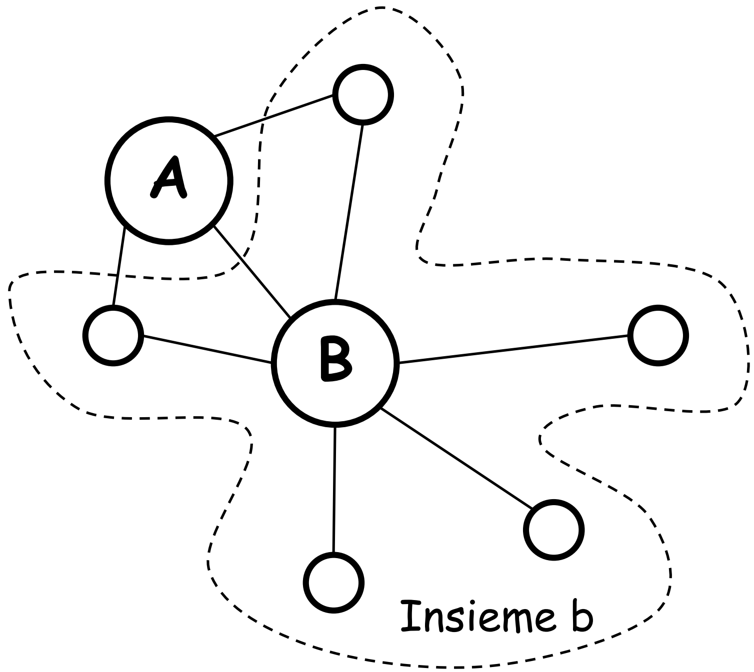
- **Quesito a**
  - $D_x(w) = c(x,w) = 2$
  - $D_x(y) = c(x,w) + c(w,y) = 4$
  - $D_x(u) = c(x,w) + c(w,u) = 7$
- **Quesito b**
  - Se il nuovo valore di c(x,y) è **c(x,y) >= 1**, $D_x(u)$ rimane uguale a 7, il nodo x non informerà i suoi vicini
  - If **c(x,y) = p_1 < 1**, il cammino minimale verso **y** avrà costo uguale **p_1+6**. e il next hop sarà il nodo **y**, **quindi x** informerà i suoi vicini della variazione di costo
  - Se **c(x,w) = $p_2 \le$ 1**, il cammino minimale verso **u** continuerà a passare attraverso **w** e il suo costo sarà uguale a **5 + p_2**; **x** informerà i suoi vicini della variazione di costo.
  - If **c(x,w) = p_3 > 6**, il cammino minimale verso **u** passerà attraverso **y** ed il suo costo sarà 11; **x** informerà i suoi vicini di questo nuovo cammino
- **Quesito c**
  - Qualsiasi variazione del costo **c(x,y)** (posto che **c(x,y) >=1**) non causerà il cambiamento del cammino minimale verso **u** e quindi **x** non emetterà nessun messaggio

Esercizio 5 (1)

- Si consideri una topologia di rete generica e la versione sincrona dell'algoritmo distance vector
- In ciascuna iterazione, un nodo invia il proprio vettore delle distanze ai suoi vicini
- Si assuma che inizialmente ciascun nodo conosca solo i costi dei rami verso i suoi vicini
- Si determini
  - qual è il massimo numero di iterazioni prima che l'algoritmo converga
  - una stima del tempo massimo di convergenza

Esercizio 5 (2)

- L'ipotesi di funzionamento sincrono comporta che, in un passo, tutti i nodi calcolino le loro tabelle delle distanze e le trasmettano ai nodi vicini
- Si consideri un generico nodo (nodo "*A*") e si osservi che in ogni iterazione:
  - il generico nodo "*A*" scambia i "distance vector" con un nodo adiacente (nodo "B");
  - a sua volta il nodo "B" scambia il vettore delle distanze con i propri vicini (*insieme b*) che saranno nodi a distanza 1 o 2 da "*A*"
- Di conseguenza dopo la prima iterazione i nodi dell'insieme b conosceranno i cammini a costo minimo di 1 o 2 hop verso il nodo "*A*"



Esercizio 5 (3)

- Sia **d** il diametro della rete, ovvero la lunghezza più elevata tra i cammini minimi esistenti tra una qualsiasi coppia di nodi della rete
- In base al ragionamento precedente, dopo ***d-1*** iterazioni tutti i nodi della rete conosceranno il costo minimo dei cammini composti da **d** hop verso tutti gli altri i nodi della rete
- Poichè **d** è la lunghezza massima di un cammino di rete, l'algoritmo convergerà dopo un numero massimo di ***d-1*** iterazioni

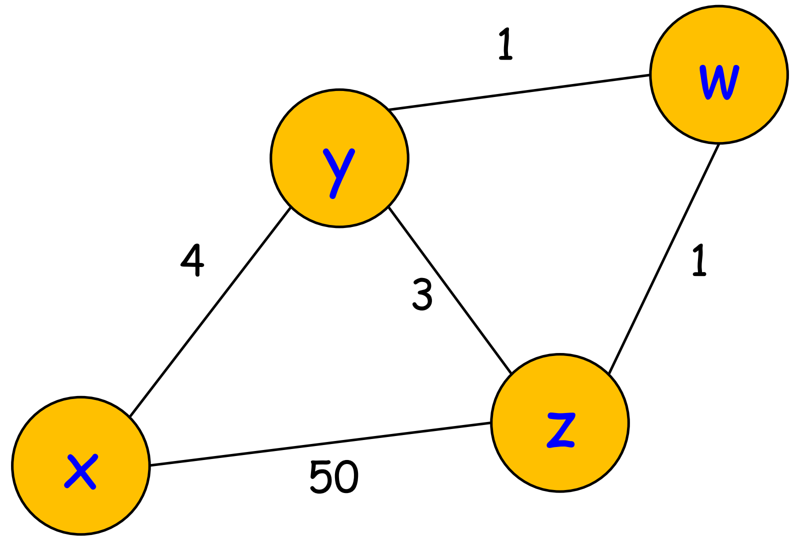
Esercizio 5 (4)

- Supponiamo che al tempo t=0 avvenga una variazione di stato in un ramo uscente dal nodo "*A*"
- Supponiamo inoltre che
  - la lunghezza massima dei rami della rete sia L_{max}=**L** km
  - il massimo tempo di trasmissione di un distance vector sia $t_{t,max}$ secondi
  - il tempo massimo di elaborazione necessario ad un nodo per il calcolo del proprio distance vector sia $t_{e,max}$ secondi
- Poichè il numero massimo di iterazioni è uguale a ***d-1***, il tempo massimo di convergenza ($T_{conv,max}$) sarà

$$T_{conv,max} = (d-1)\cdot\left(t_{t,max} + t_{e,max} + 5\cdot L10^{-6}\right)$$

Esercizio 6 (1)

- Si consideri la rete mostrata in figura
- I costi associati ai link sono:
  - c(x,y)=4
  - c(x,z)=50
  - c(y,w)=1
  - c(z,w)=1
  - c(y,z)=3
- Si assuma che sia usata l'inversione avvelenata (poisoned reverse) nell'algoritmo di instradamento distance vector



Esercizio 6 (2)

- Quesito (a)
  - Quando l'algoritmo è stabilizzato, qual è il valore delle distanze da x riportata nei distance vector che i router z, y e w si scambiano ?
- Quesito (b)
- Si assuma che al tempo t0 il costo del collegamento tra x e y assuma il valore 60
  - Si verificherà il problema del conteggio all'infinito ?
  - Quante iterazioni sono necessarie perché l'instradamento raggiunga uno stato stazionario ?

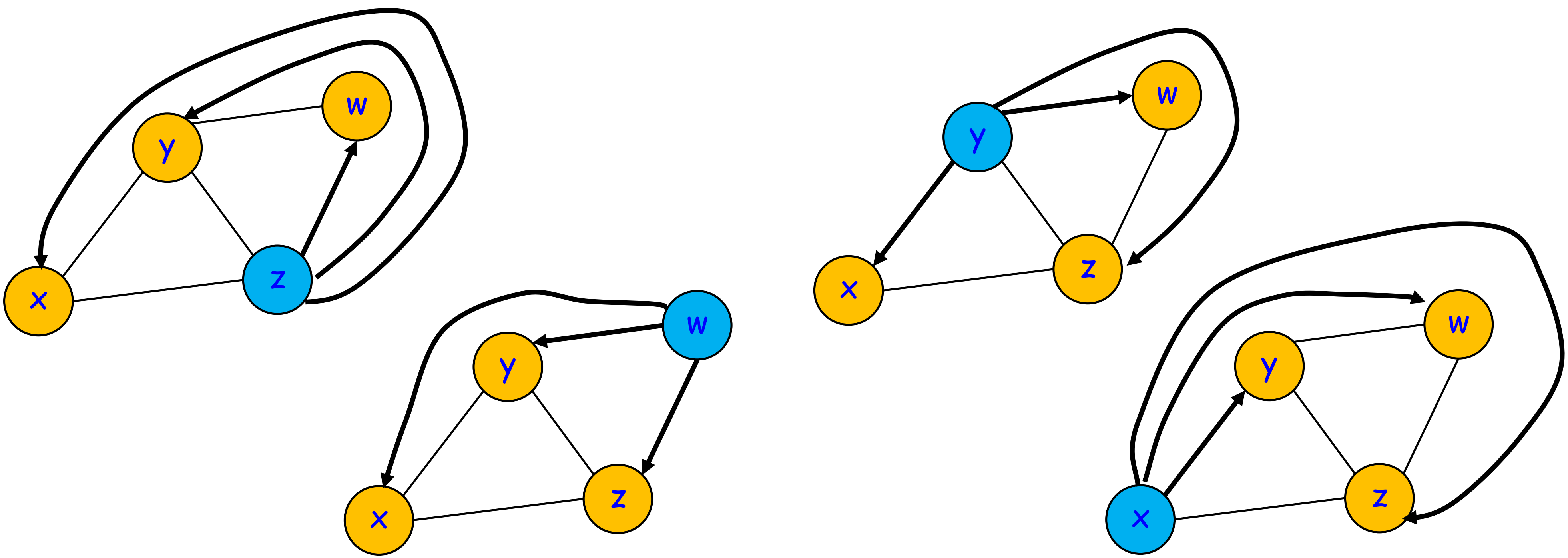
Esercizio 6 (3)

- Quesito (a)
  - Quando l'algoritmo ha raggiunto lo stato stabile, i vettori delle distanze sono i seguenti:

		Nodo destinazione			
		Z	W	Y	X
Nodo origine	Z	---	1	2	6
	W	1	---	1	5
	Y	2	1	---	4
	X	6	5	4	---

Esercizio 6 (4)

- Quesito (a)
  - Nello stato stabile gli instradamenti sono i seguenti



Esercizio 6 (5)

- Quesito (a)
  - La regola dell'inversione avvelenata prescrive che "se un nodo α, per raggiungere γ, instrada i pacchetti verso β, il nodo α comunica a β che la sua distanza verso γ è infinita (in questo modo β non instraderà i pacchetti per γ verso α)"
  - Di conseguenza le distanze dal nodo x annunciate dai nodi z, w e y sono le seguenti:

		Nodo destinazione		
		Z	W	Y
Nodo origine	Z		$D_z(x)=\infty$	$D_z(x)=6$
	W	$D_w(x)=5$		$D_w(x)=\infty$
	Y	$D_y(x)=4$	$D_y(x)=4$	

Esercizio 6 (6)

- Quesito (b)
  - L'inversione avvelenata non evita il conteggio all'infinito
  - Infatti, il processo di convergenza è mostrato nelle tabelle seguenti
    - al tempo t1, y aggiorna il suo distance vector e informa i nodi w e z
    - ai tempi t1, t2, t3, t4 proseguono le iterazioni ed il costo dei cammini cresce e tra i nodi w, y, z si forma un loop nel calcolo dei costi verso il router x

Tempo	t0	t1	t2	t3	t4
z	$D_z^w(x)=\infty$ $D_z^y(x)=6$		Nessuna variazione	$D_z^w(x)=\infty$ $D_z^y(x)=11$	
w	$D_w^y(x)=\infty$ $D_w^z(x)=5$		$D_w^y(x)=\infty$ $D_w^z(x)=10$		Nessuna variazione
y	$D_y^w(x)=4$ $D_y^z(x)=4$	$D_y^w(x)=9$ $D_y^z(x)=\infty$		Nessuna variazione	$D_y^w(x)=14$ $D_y^z(x)=\infty$

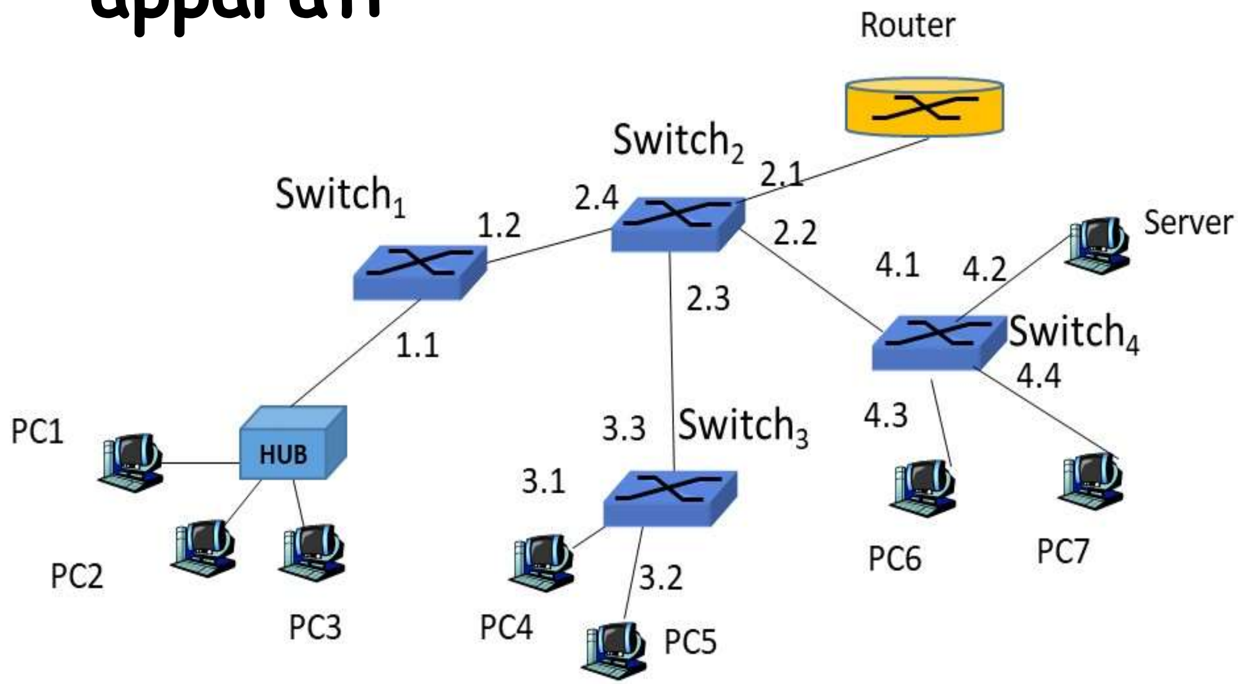
Esercizio 6 (7)

- Continuando le iterazioni
  - al tempo t1, y aggiorna il suo distance vector e informa i nodi w e z
  - al tempo t27, z rivela che il costo minimo verso x è 50, attraverso il link diretto
  - al tempo t29, w apprende che il costo minimo verso x è 51 attraverso z
  - al tempo t30, y aggiorna a 52 il suo costo minimo verso x (verso w)
  - Infine, al tempo t31, il routing si stabilizza

Tempo	t27	t28	t29	t30	t31
z	$D_z^w(x)=50$ $D_z^y(x)=50$				$D_z^w(x)=\infty$ $D_z^y(x)=55$ $D_z^z(x)=50$
w		$D_w^y(x)=\infty$ $D_w^z(x)=50$	$D_w^y(x)=51$ $D_w^z(x)=\infty$		$D_w^w(x)=\infty$ $D_w^y(x)=\infty$ $D_w^z(x)=51$
y		$D_y^w(x)=53$ $D_y^z(x)=\infty$		$D_y^w(x)=\infty$ $D_y^z(x)=52$	$D_y^w(x)=52$ $D_y^y(x)=60$ $D_y^z(x)=53$

Esercizio 7 (1)

- **Si consideri la LAN Ethernet mostrata in figura in cui sono riportati anche gli indirizzi MAC degli apparati**



Nome	Indirizzo MAC
PC1	00-CC-AA-11-22-01
PC2	00-CC-AA-11-22-02
PC3	00-CC-AA-11-22-03
PC4	00-CC-AA-11-22-04
PC5	00-CC-AA-11-22-05
PC6	00-CC-AA-11-22-06
PC7	00-CC-AA-11-22-07
Server	00-CC-AA-11-22-0A
Router	00-CC-AA-11-22-0B

Esercizio 7 (2)

- (a) Ipotizzando che la lunghezza di tutti i segmenti di collegamento siano uguali a D=400 metri, determinare il limite teorico (Lmin) della dimensione minima della trama (si supponga un ritardo di propagazione δ=5 μs/km ed un ritmo di trasmissione delle trame uguale a R=40 Mbit/s)

Esercizio 7 (3)

- (b) Supponendo che, all'istante t, le tabelle di inoltro (forwarding table) degli switch siano:

Switch1			Switch2	
Indirizzo	Interfaccia		Indirizzo	Interfaccia
00-CC-AA-11-22-03	1.1		00-CC-AA-11-22-03	2.4
00-CC-AA-11-22-04	1.2		00-CC-AA-11-22-04	2.3

  - Indicare l'insieme dei collegamenti sui quali vengono inoltrate le trame inviate dal terminale avente indirizzo MAC SOURCE_MAC_ADD e dirette verso il terminale con indirizzo MAC DEST_MAC_ADD

Trama 1

SOURCE_MAC_ADD	00-CC-AA-11-22-0A
DEST_MAC_ADD	00-CC-AA-11-22-04

Trama 2

SOURCE_MAC_ADD	00-CC-AA-11-22-02
DEST_MAC_ADD	00-CC-AA-11-22-0B

- (c) Riempire, se necessario, le altre righe delle tabelle di switch dopo che siano inoltrate le due trame 1 e 2

Esercizio 7 (4)

- **Quesito (a)**
  - La lunghezza minima L_{min} di una trama è data da

$$\frac{L_{\min}}{C} \geq 2 \cdot t_{p\_max}$$

    - dove t_{p_max} è il massimo tempo di ritardo tra due terminali nello stesso dominio di collisione
    - La massima distanza tra due terminali è d_{max}=800 m
      - da cui

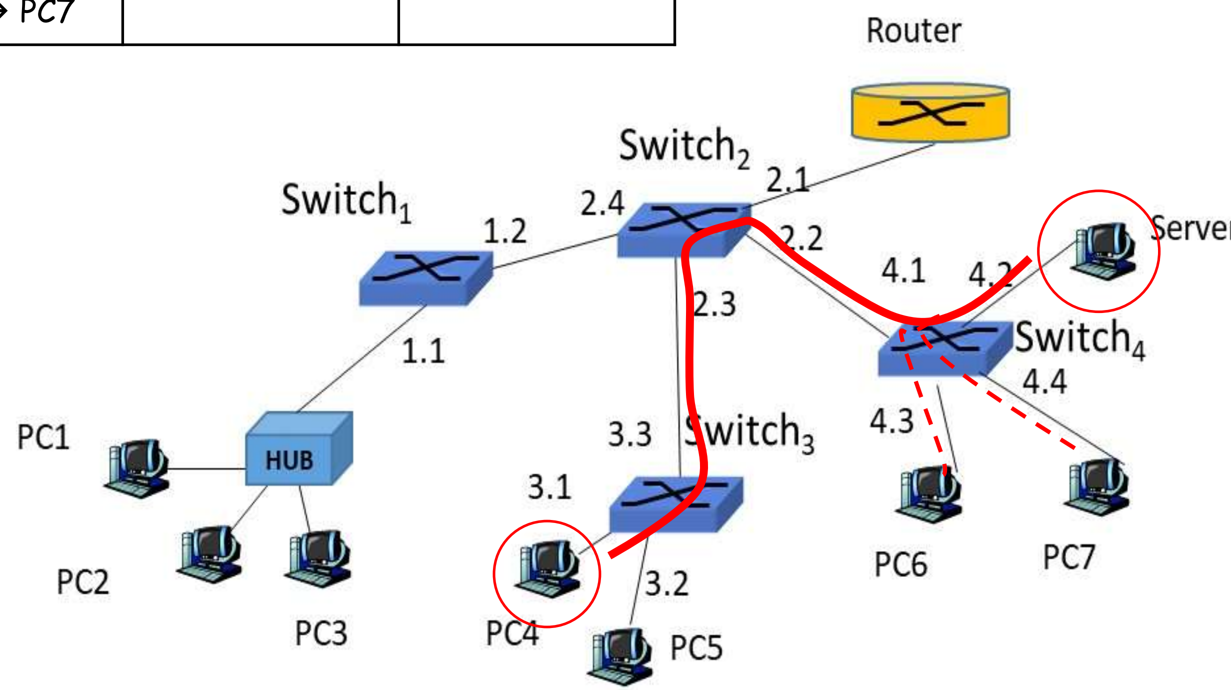
$$L_{min} \geq 2 \cdot C \cdot t_{p\_max} = 2 \cdot C \cdot 0{,}8 \cdot 5 \cdot 10^{-6}$$

    - quindi
  - L_{min}=320 bit

Esercizio 7 (5)

- **Quesito (b): percorso trama 1: da Server → PC4**

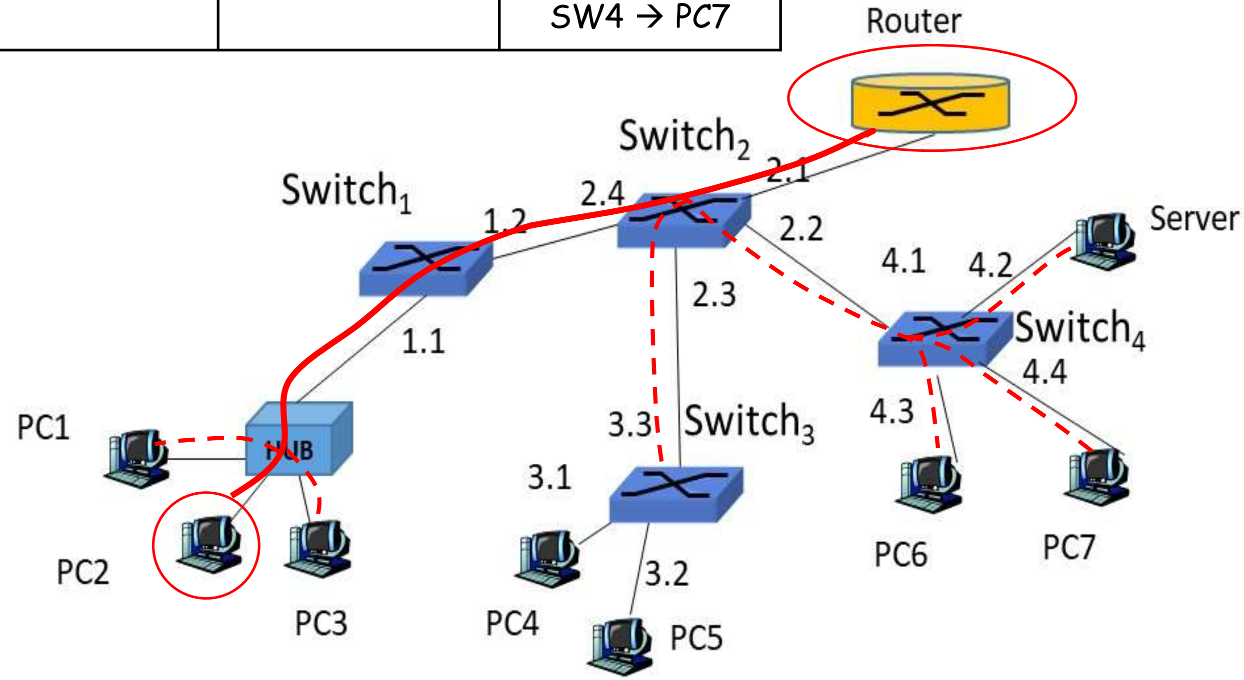
OA → SW4	SW4 → SW2	SW2 → SW3	SW3 → 04
	SW4 → PC6		
	SW4 → PC7		



Esercizio 7 (6)

- **Quesito (b): percorso trama 2: da PC2 → Router**

PC2 → SW1	SW4 → SW2	SW2 → Router		
PC2 → PC1		SW2 → SW3	SW3 → PC4	SW4 → Server
PC2 → PC3		SW2 → SW4	SW3 → PC5	SW4 → PC6
				SW4 → PC7



Esercizio 7 (7)

- **Quesito (c)**
  - Le tabelle di forwarding degli switch si modificano in questo modo:

Switch1			Switch2	
Indirizzo	Interfaccia		Indirizzo	Interfaccia
00-CC-AA-11-22-03	1.1		00-CC-AA-11-22-03	2.4
00-CC-AA-11-22-04	1.2		00-CC-AA-11-22-04	2.3
00-CC-AA-11-22-02	1.1		00-CC-AA-11-22-0A	2,2
			00-CC-AA-11-22-02	2.1

Switch3			Switch4	
Indirizzo	Interfaccia		Indirizzo	Interfaccia
00-CC-AA-11-22-0A	3.3		00-CC-AA-11-22-0A	4.2
00-CC-AA-11-22-02	3.3		00-CC-AA-11-22-02	4.1