

Internet and Data Centers

algoritmi distance vector

G. Di Battista, M. Patrignani

copyright notice

- all the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as “material”) are protected by copyright
- this material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide
- this material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes
- any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement
- this copyright notice must always be redistributed together with the material, or its portions

algoritmi distance vector

- ogni is invia una tabella, detta *distance vector*, ad ogni is adiacente
- il distance vector contiene la parte *essenziale* della tabella di instradamento dell'is
 - vengono omessi dettagli locali, come i nomi delle interfacce ecc.
- gli is ricalcolano le tabelle di instradamento integrando le indicazioni presenti nei vari distance vector ricevuti

algoritmi distance vector

- le destinazioni vengono apprese tramite un processo di passa-parola
 - inizialmente un is conosce solo le destinazioni ad esso direttamente connesse
 - nel ricevere un distance vector l'is somma al costo di ogni destinazione contenuta nel distance vector il costo della linea di ingresso
 - ogni destinazione della tabella di instradamento è corredata dal costo del cammino dall'is alla destinazione stessa
- esempio di priorità nelle metriche sui cammini
 - prima considera il costo, a parità di costo il numero di hop, a parità di costo e di hop fai una scelta random

algoritmi distance vector

- invio dei distance vector (alternative)
 - ad ogni modifica della tabella di instradamento
 - periodicamente
- i nodi adiacenti da aggiornare
 - vengono appresi tramite protocolli appositi
 - sono ignoti
 - i distance vector vengono inviati in multicast
- lentezza nella convergenza
 - difficile da usare per reti con più di 1.000 nodi

esempio di distance vector

distance vector pervenuti sulle linee

linea 1		linea 2		linea 3		linea 4	
net	cost	net	cost	net	cost	net	cost
A	0	A	3	A	5	C	5
B	3	B	2	B	0	D	7
D	2	C	6	C	2	E	2

sommati i costi delle linee

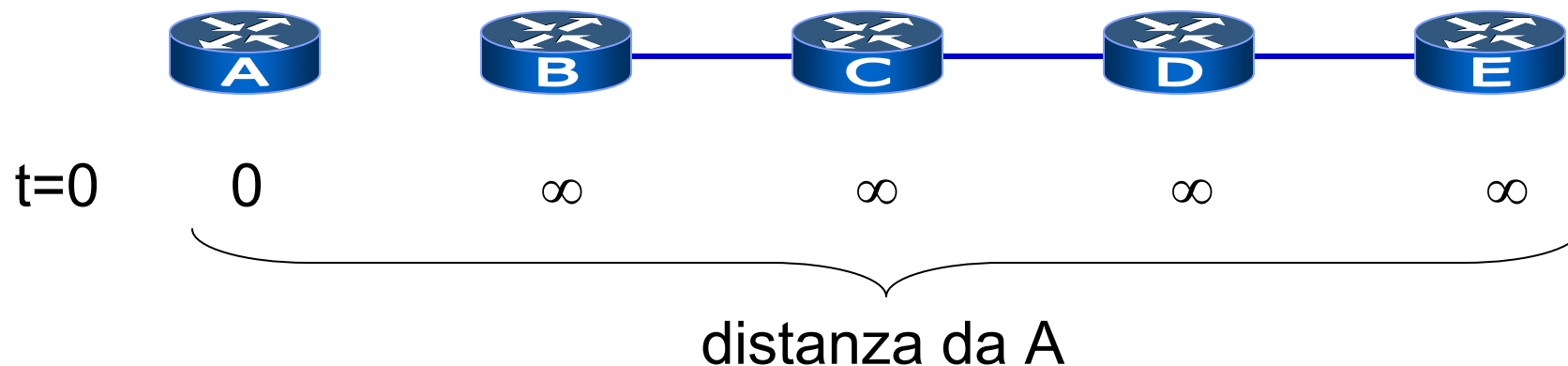
costo 2		costo 2		costo 3		costo 1	
net	cost	net	cost	net	cost	net	cost
A	2	A	5	A	8	C	6
B	5	B	4	B	3	D	8
D	4	C	8	C	5	E	3

tabella risultante

net	line	cost
A	1	2
B	3	3
C	3	5
D	1	4
E	4	3


il problema del count-to-infinity

- gli algoritmi distance vector reagiscono rapidamente alle “buone notizie” e lentamente alle “cattive notizie”
- esempio di “buona notizia”
 - supponiamo che all'istante iniziale la linea tra il router A e il router B sia guasta e supponiamo che la metrica sia relativa al numero di hop
 - condizioni iniziali



il problema del count-to-infinity

- supponiamo che linea tra A e B venga riparata

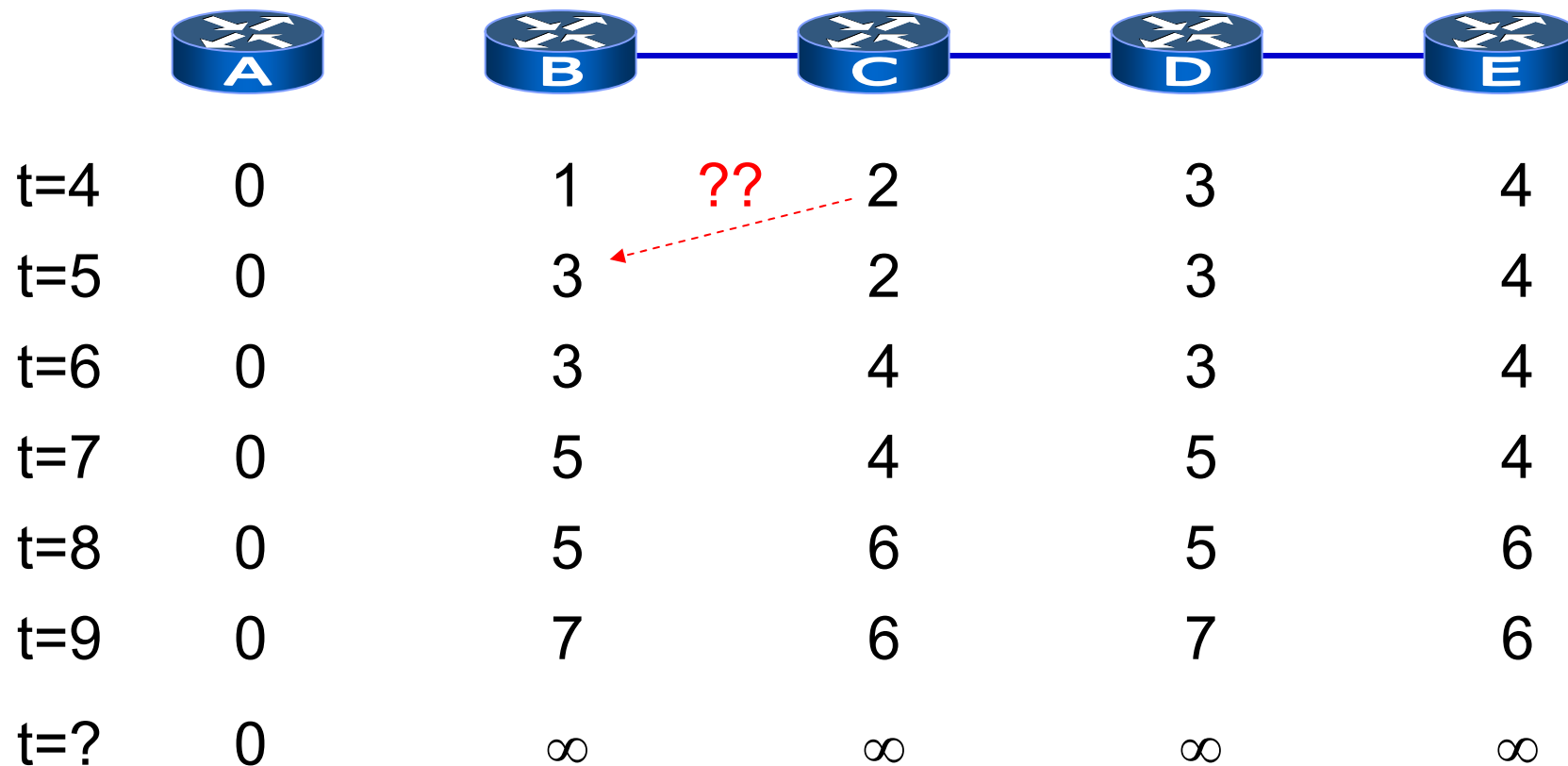


t=0	0	∞	∞	∞	∞
t=1	0	1	∞	∞	∞
t=2	0	1	2	∞	∞
t=3	0	1	2	3	∞
t=4	0	1	2	3	4

- la "buona notizia" si propaga nelle tabelle d'instradamento dei vari router

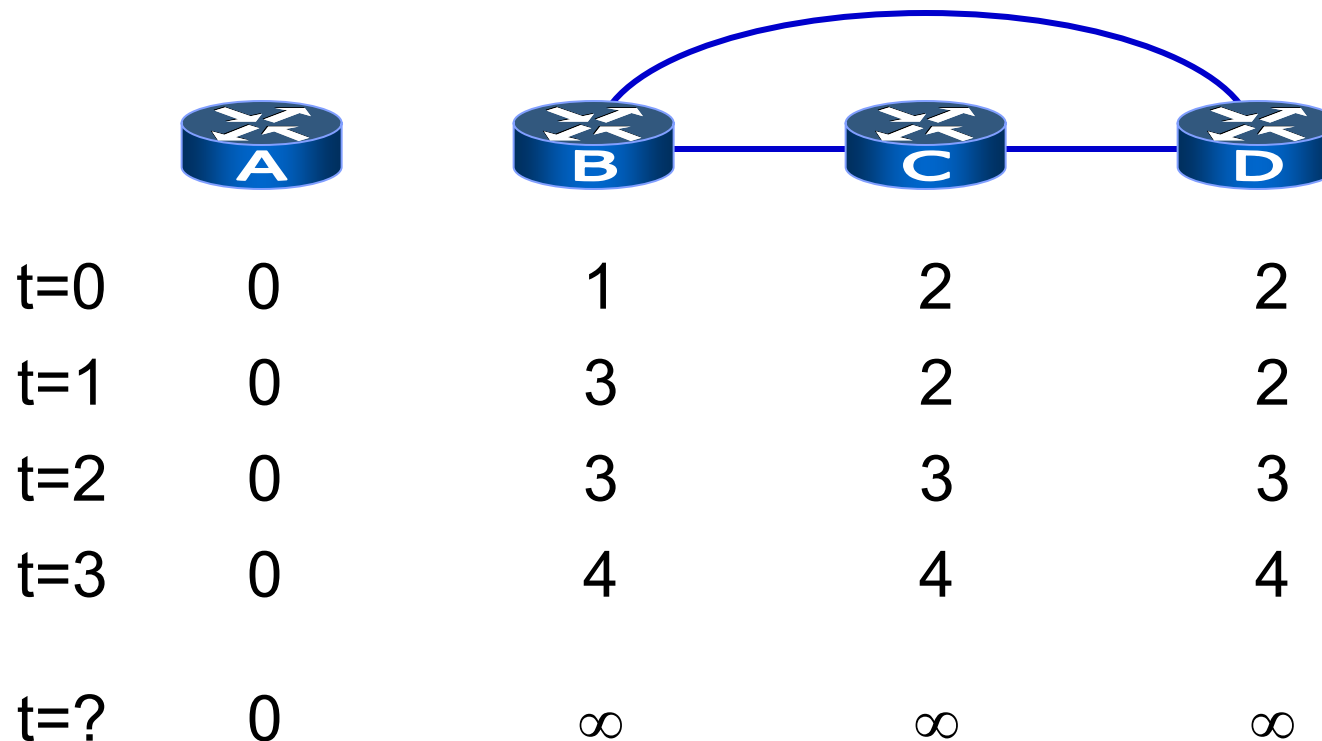
il problema del count-to-infinity

- esempio di “cattiva notizia”
 - supponiamo che la linea tra A e B vada nuovamente fuori servizio



il problema del count-to-infinity

- altro esempio di “cattiva notizia”
 - supponiamo che la linea tra A e B vada fuori servizio

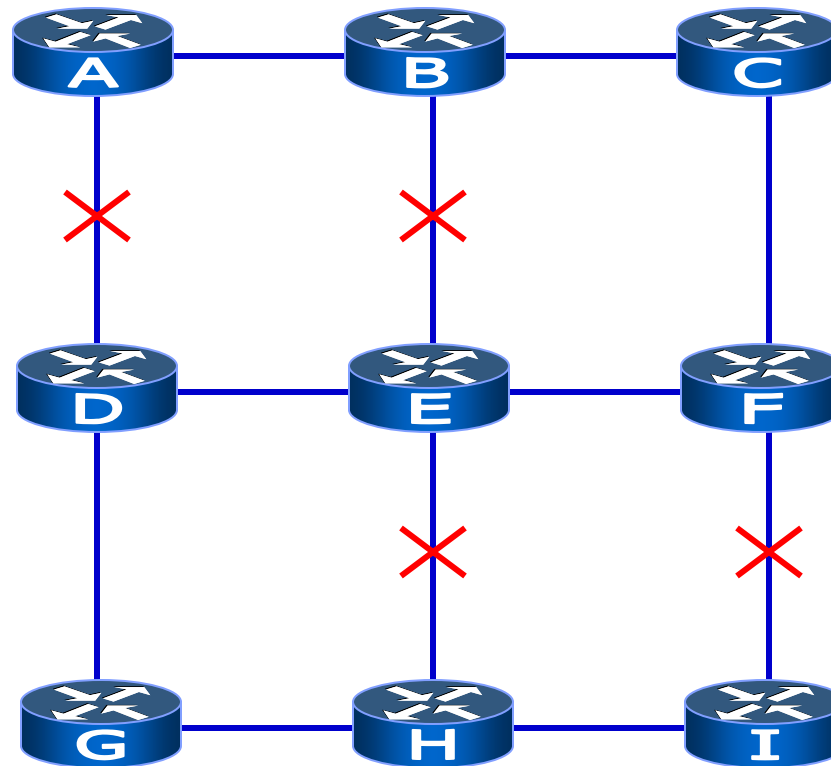


il problema del count-to-infinity

- una buona notizia arriva a distanza k in k passi
- le cattive notizie si propagano in un tempo che è funzione del valore convenzionale attribuito ad *infinito*
- per questo motivo si attribuisce normalmente ad *infinito* il valore della lunghezza del cammino più lungo più 1

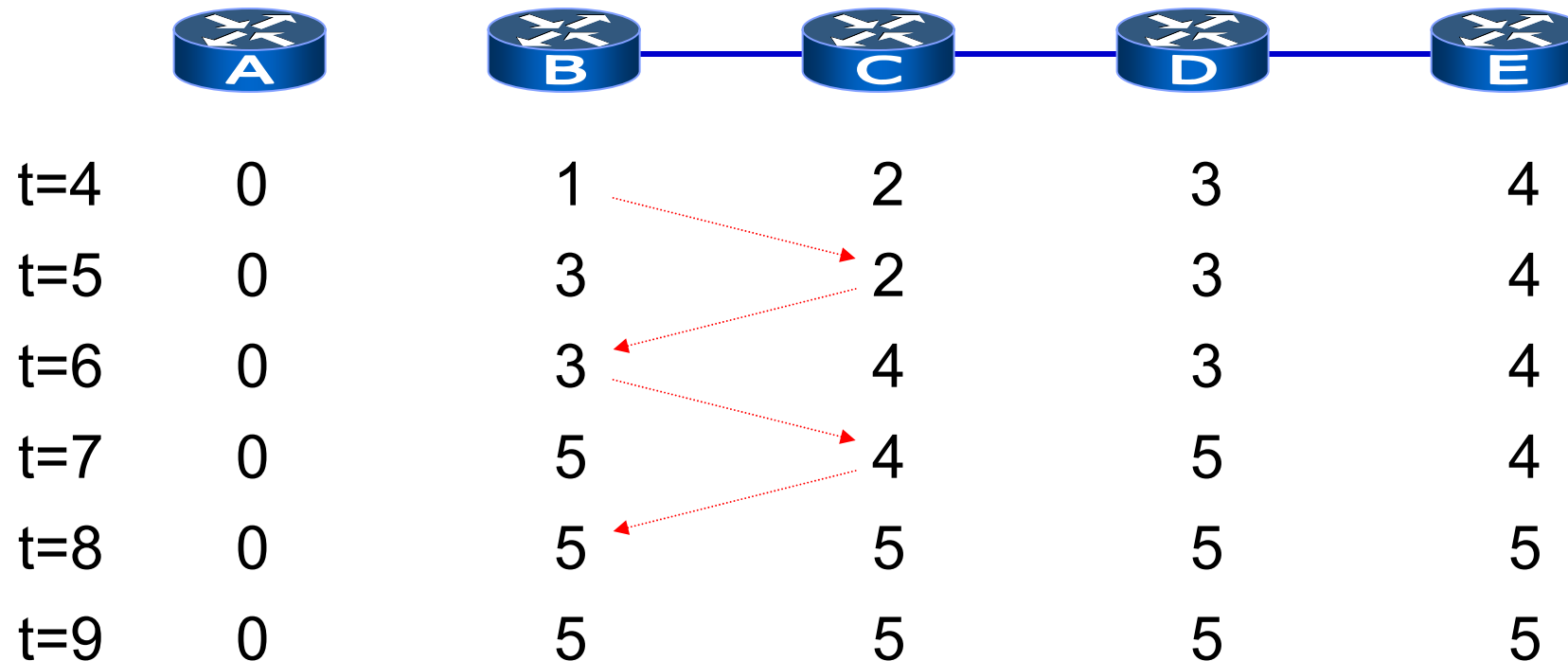
valore convenzionale di infinito

- qual è il valore ottimale di infinito per la rete rappresentata qui sotto?



valore convenzionale di infinito

- attribuiamo ad infinito il valore convenzionale di cinque
 - supponiamo che la linea tra A e B vada fuori servizio



- la cattiva notizia si propaga in 4 passi (infinito – 1)

reazione alle cattive notizie

- se la rete viene disconnessa
 - si incorre nel problema del count-to-infinity
 - occorre definire un valore convenzionale di infinito
- se la rete rimane connessa
 - è possibile dimostrare (modello di Bellman e Ford) che l'algoritmo converge sempre ad una soluzione ottima
 - indipendentemente dai valori iniziali delle distanze negli is
 - il numero di passi è lineare se la metrica è il numero di hop

efficienza degli algoritmi distance vector

- ci sono vari modi per valutare l'efficienza di un algoritmo di instradamento
 - valutare il tempo impiegato da ciascun router per portare la propria tabella ad una situazione stabile
 - contare il numero di passi che l'algoritmo impiega a convergere, immaginando che gli eventi siano sincronizzati
 - contare il numero di pacchetti che vengono scambiati sulla rete per convergere

dv – efficienza (numero di passi)

- supponiamo che
 - nella rete ci siano n nodi ed m link
 - la metrica sia relativa solo al numero di hop
- tra i nodi più lontani ci sono al massimo $n-1$ link (lunghezza massima del cammino più lungo)
 - una buona notizia si propaga su tutta la rete in al più $n-1$ passi
 - se attribuiamo a infinito il valore n allora una cattiva notizia si propaga su tutta la rete in al più $n-1$ passi

dv – efficienza (lavoro svolto dai router)

- in ciascuno dei passi ogni router scandisce al più m distance vector
 - infatti m è anche il numero massimo di linee intorno a un router
- ogni tabella ha al più $O(n)$ righe (entries)
- in ciascuno dei passi ogni router spende tempo $O(nm)$
- dato che la convergenza richiede nel caso peggiore $O(n)$ passi, ogni router calcola per un tempo $O(n^2m)$