

Számítógépes Hálózatok

7a. Előadás: Hálózati réteg

Based on slides from **Zoltán Ács ELTE** and D. Choffnes Northeastern U., Philippa Gill from StonyBrook University , Revised Spring 2016 by S. Laki

Távolságvektor alapú forgalomirányítás

2

- Dinamikus algoritmusoknak 2 csoportja van:
 - ▣ távolságvektor alapú illetve (distance vector routing)
 - ▣ kapcsolatállapot alapú (link-state routing)

- **Távolságvektor alapú:** Minden router-nek egy táblázatot kell karbantartania, amelyben minden célhoz szerepel a legrövidebb ismert távolság, és annak a vonalnak az azonosítója, amelyiken a célhoz lehet eljutni. A táblázatok a szomszédoktól származó információk alapján frissítik.
 - ▣ Elosztott Bellman-Ford forgalomirányítási algoritmusként is nevezik.
 - ▣ ARPANET eredeti forgalomirányító algoritmus ez volt. RIP (Routing Information Protocol) néven is ezt használták.

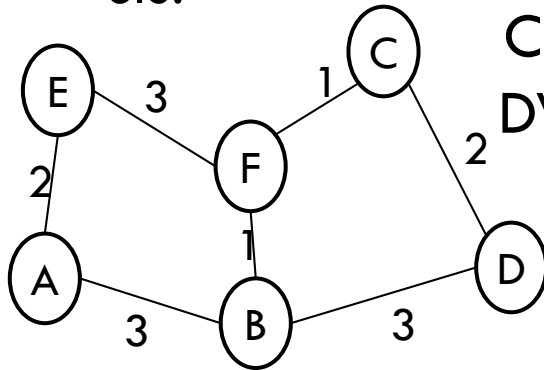
Távolságvektor alapú forgalomirányítás

Elosztott Bellman-Ford algoritmus

3

KÖRNYEZET ÉS MŰKÖDÉS

- Minden csomópont csak a közvetlen szomszédjaival kommunikálhat.
- Aszinkron működés.
- Minden állomásnak van saját távolság vektora. Ezt periodikusan elküldi a direkt szomszédoknak.
- A kapott távolság vektorok alapján minden csomópont új táblázatot állít elő.



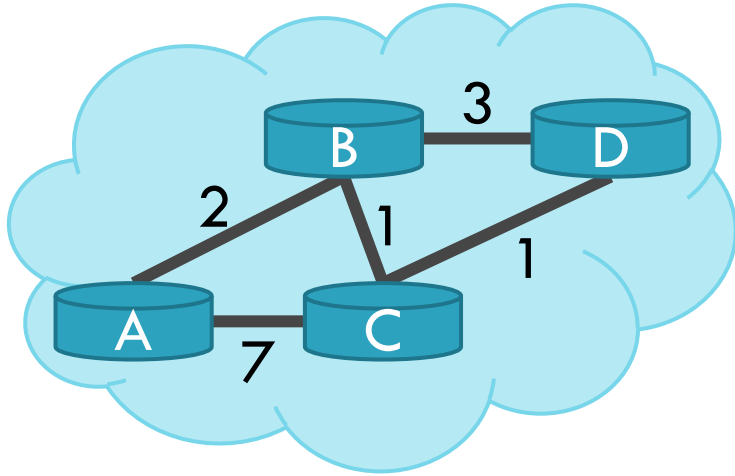
C állomás
DV táblája

Cél	Ktsz.
A	5
B	2
D	2
E	4
F	1

- Nincs bejegyzés C-hez
- Kezdetben csak a közvetlen szomszédokhoz van info
 - ▣ Más célállomások költsége = ∞
- Végül kitöltött vektort kapunk

Distance Vector Initialization

4



Node A

Dest.	Cost	Next
B	2	B
C	7	C
D	∞	

Node B

Dest.	Cost	Next
A	2	A
C	1	C
D	3	D

Node C

Dest.	Cost	Next
A	7	A
B	1	B
D	1	D

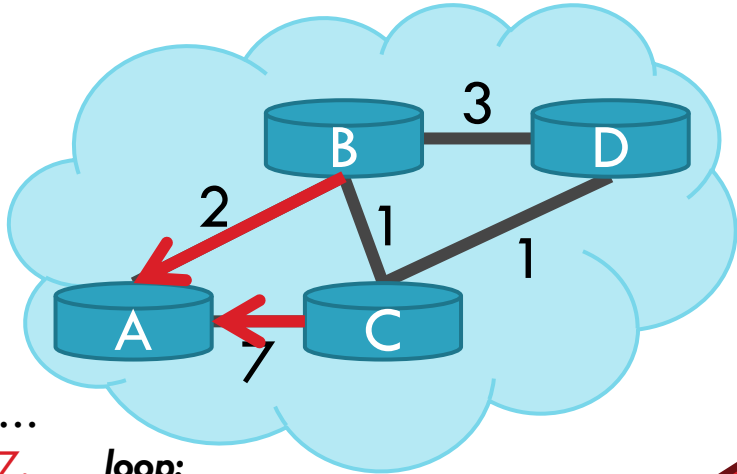
Node D

Dest.	Cost	Next
A	∞	
B	3	B
C	1	C

1. **Initialization:**
2. **for all** neighbors V **do**
3. **if** V adjacent to A
4. $D(A, V) = c(A, V);$
5. **else**
6. $D(A, V) = \infty;$
- ...

Distance Vector: 1st Iteration

5



Node A

Dest.	Cost	Next
B	2	B
C	3	B
D	5	B

Node B

Dest.	Cost	Next
A	2	A
C	1	C
D	2	C



$D(A,C)$

$D(A,C), D(A,B) + D(B,C)$

$D(A,D) = \min(D(A,D), D(A,B) + D(B,D))$
 $= \min(8, 3 + 3) = 5$

4

Next

B

B

C

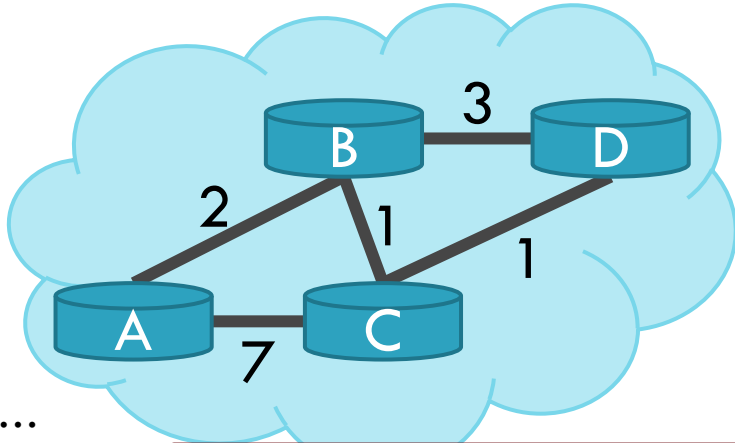
C

```

...
7. loop:
...
12. else if (update D(V, Y) received from neighbor V)
13.   for all destinations Y
14.     if (destination Y is not the neighbor V)
15.       D(A, Y) = min(D(A, Y), D(V, Y) + D(V, Y));
16.     else
17.       D(A, Y) = min(D(A, Y), D(V, Y) + D(V, Y));
18.   if (there is a new min. for dest. Y)
19.     send D(A, Y) to all neighbors
20. forever
  
```

Distance Vector: End of 3rd Iteration

6



Node A

Dest.	Cost	Next
B	2	B
C	3	B
D	4	B

Node B

Dest.	Cost	Next
A	2	A
C	1	C
D	2	C

- Nothing changes, algorithm terminates
- Until something changes...

Dest.	Cost	Next	Dest.	Cost	Next
A	3	B	A	4	C
B	1	B	B	2	C
D	1	D	C	1	C

loop

else

for

$D(A, Y) =$

$\min(D(A, Y),$

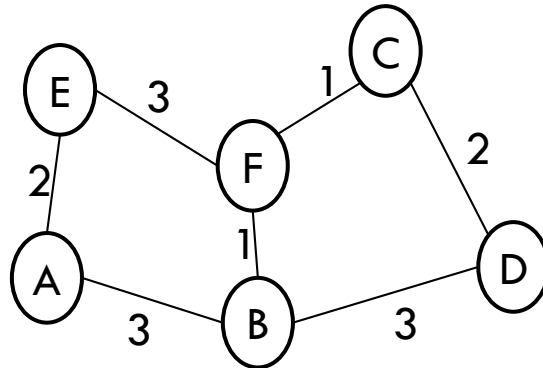
$D(A, V) + D(V, Y));$

if (there is a new min. for dest. Y)

send $D(A, Y)$ to all neighbors

forever

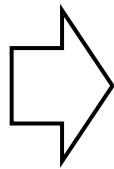
Elosztott Bellman-Ford algoritmus – példa



Becsült késleltetés A-tól kezdetben		
A	cost	N. Hop
B	3	B
C	∞	-
D	∞	-
E	2	E
F	∞	-

B vektora A-nak	
A	3
B	0
C	∞
D	3
E	∞
F	1

E vektora A-nak	
A	2
B	∞
C	∞
D	∞
E	0
F	3



Új becslét késleltetés A-tól		
A	cost	N. Hop
B	3	B
C	∞	-
D	6	B
E	2	E
F	4	B

A vektora B-nek és E-nek	
A	0
B	3
C	∞
D	6
E	2
F	4

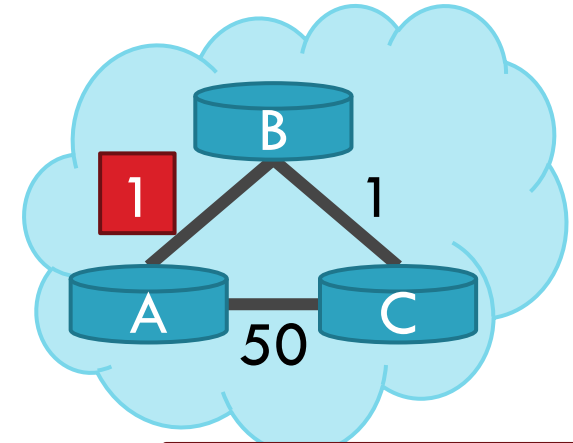


Új becslét késleltetés A-tól		
A	cost	N. Hop
B	3	B
C	5	B
D	6	B
E	2	E
F	4	B

```

7.  loop:
8.    wait (link cost update or update message)
9.    if (c(A,V) changes by d)
10.     for all destinations Y through V do
11.       D(A,Y) = D(A,Y) + d
12.     else if (update D(V, Y) received from V)
13.       for all destinations Y do
14.         if (destination Y through V)
15.           D(A,Y) = D(A,V) + D(V, Y);
16.       else
17.         D(A, Y) = min(D(A, Y), D(A, V) + D(V, Y));

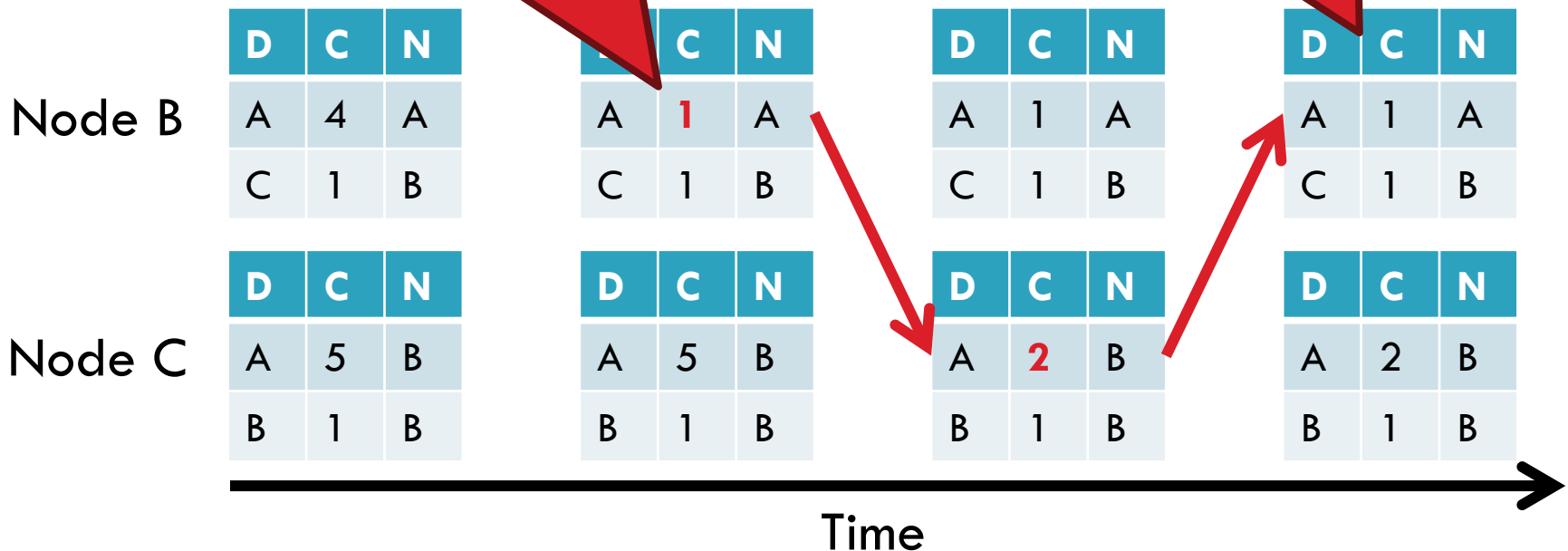
```



Link Cost
Algorithm

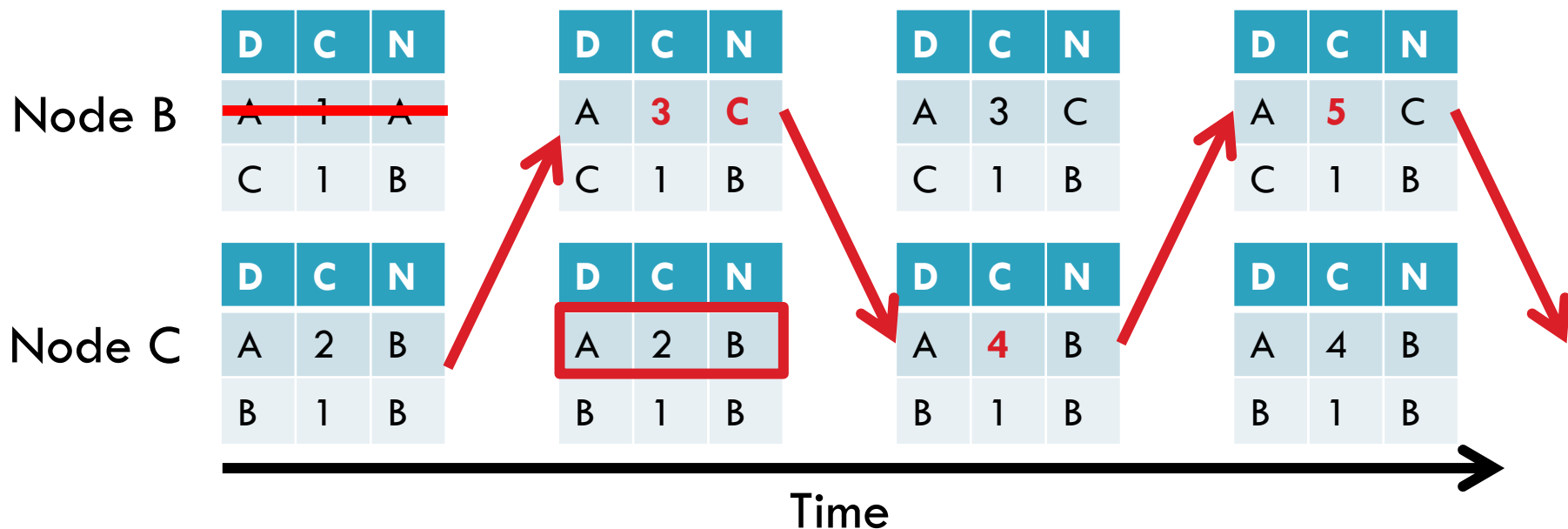
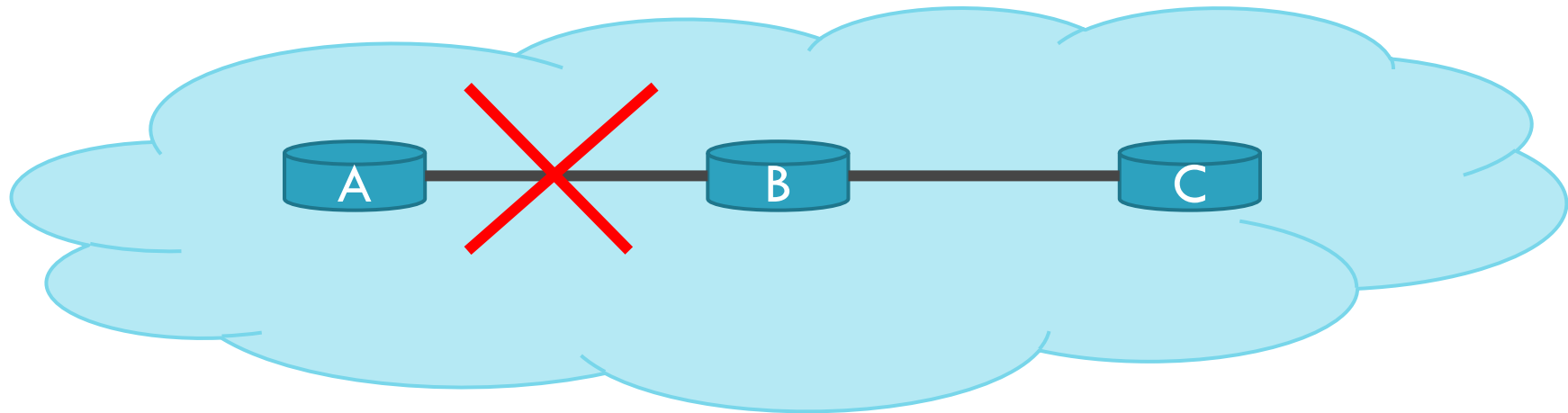
Good news travels fast

Algorithm
terminates



Távolság vektor protokoll – Végtelenig számolás problémája (count to infinity)

9

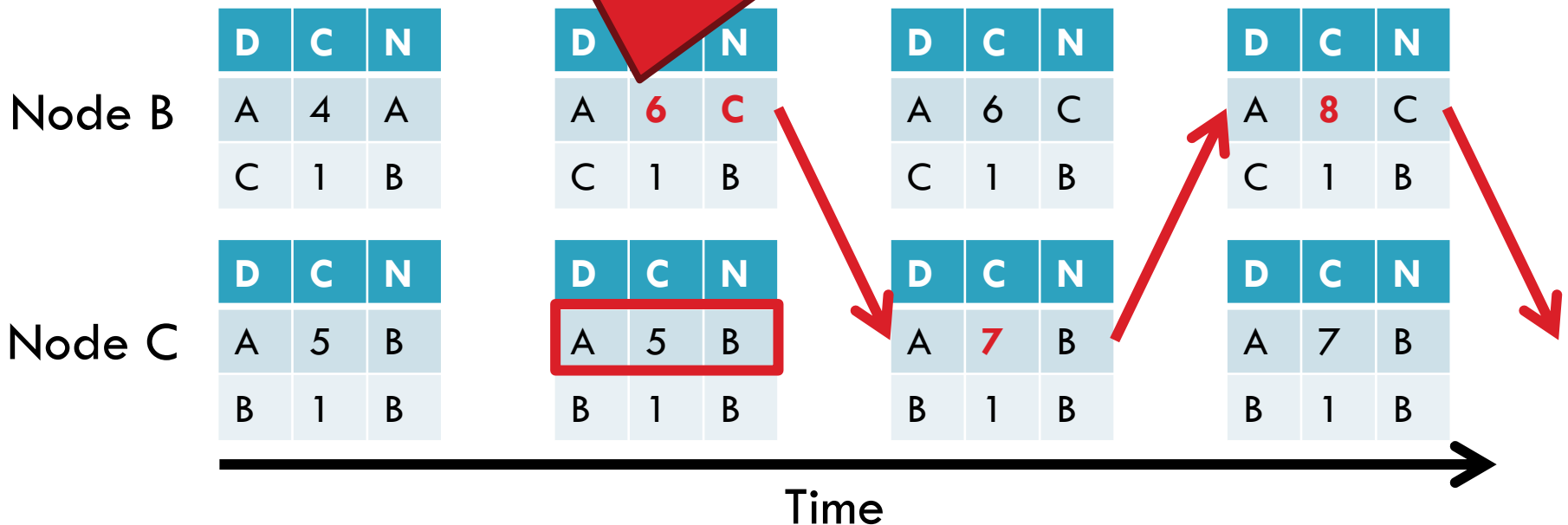
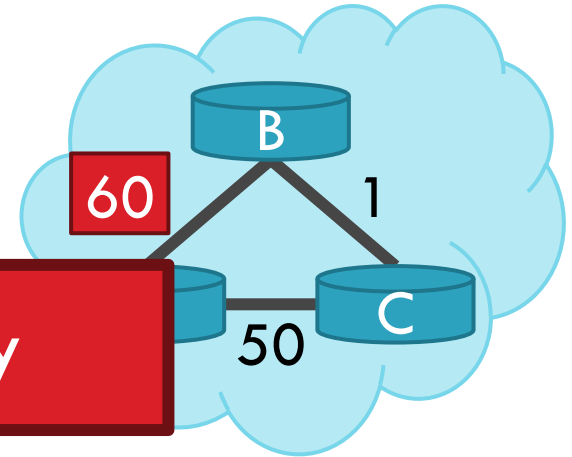


Példa - Count to Infinity Problem

10

- Node B knows $D(C, A) = 5$
- However, B does not know the path is $C \rightarrow B \rightarrow A$
- Thus, $D(B, A)$

Bad news travels slowly



Elosztott Bellman-Ford algoritmus – *Végtelenig számolás problémája*

11

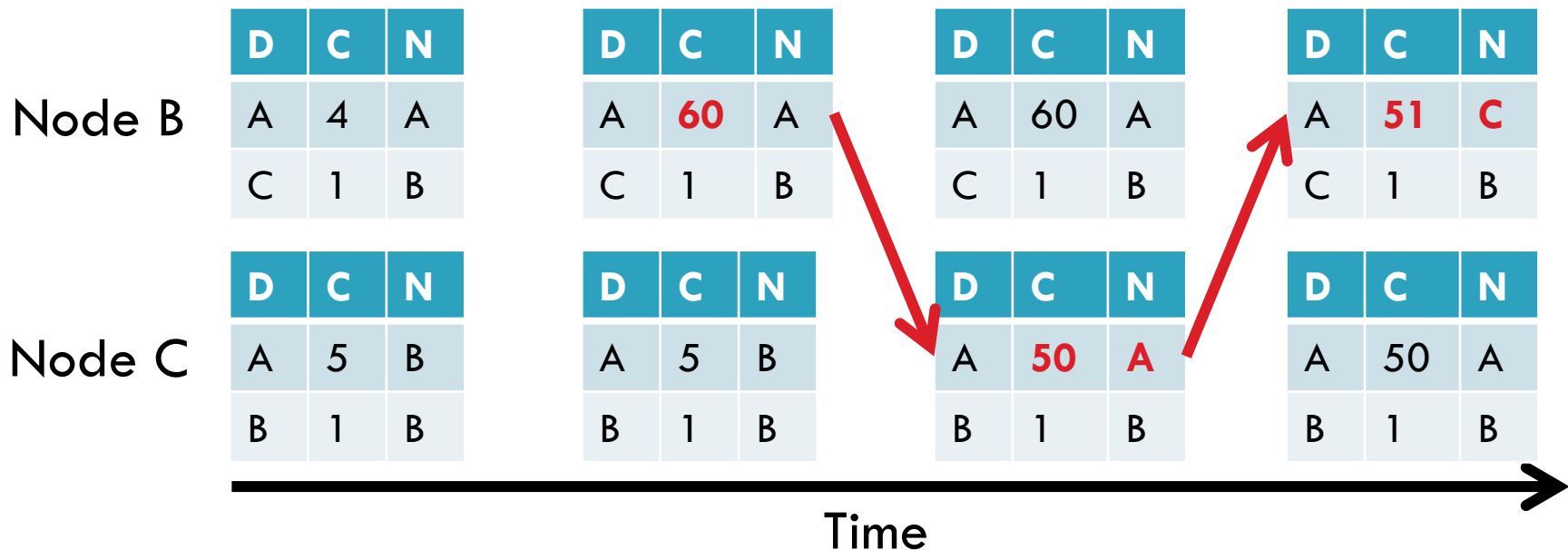
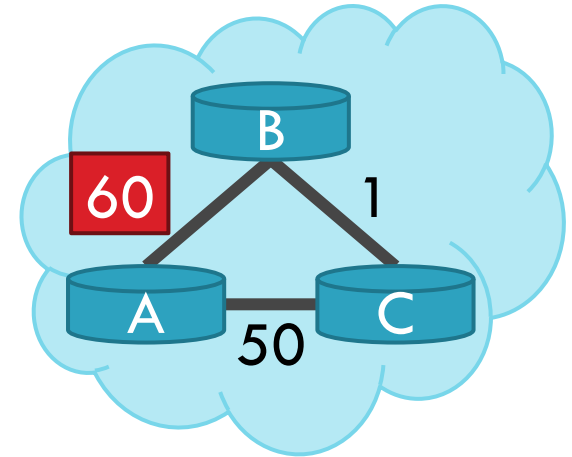
PROBLÉMA

- A „jó hír” gyorsan terjed.
- A „rossz hír” lassan terjed.
- Azaz ciklusok keletkezhetnek.
- Lehetséges megoldás:
 - ▣ **„split horizon with poisoned reverse”**: negatív információt küld vissza arról a szomszédjának, amit tőle „tanult”. (RFC 1058)

Split horizon with Poisoned Reverse

12

- Ha C B-n keresztül irányítja a forgalmat A állomáshoz
 - ▣ C állomás B-nek $D(C, A) = \infty$ távolságot küld
 - ▣ Azaz B állomás nem fog C-n keresztül irányítani az A-ba menő forgalmat



Vége



□ Köszönöm a figyelmet!