



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ - ΤΜΗΥΠ  
ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΙΙ

---

*B. Μεγαλοοικονόμου*

Κατανεμημένες Βάσεις Δεδομένων

(παρουσίαση βασισμένη εν μέρη σε σημειώσεις των Silberchatz, Korth και Sudarshan και του C. Faloutsos)



# Γενική επισκόπηση

---

- Φυσικός σχεδιασμός, Δεικτοδότηση
- Βελτιστοποίηση ερωτημάτων
- Επεξεργασία δοσοληψιών
- Αρχιτεκτονική συστημάτων βάσεων δεδομένων
  - Κατανεμημένες βάσεις δεδομένων



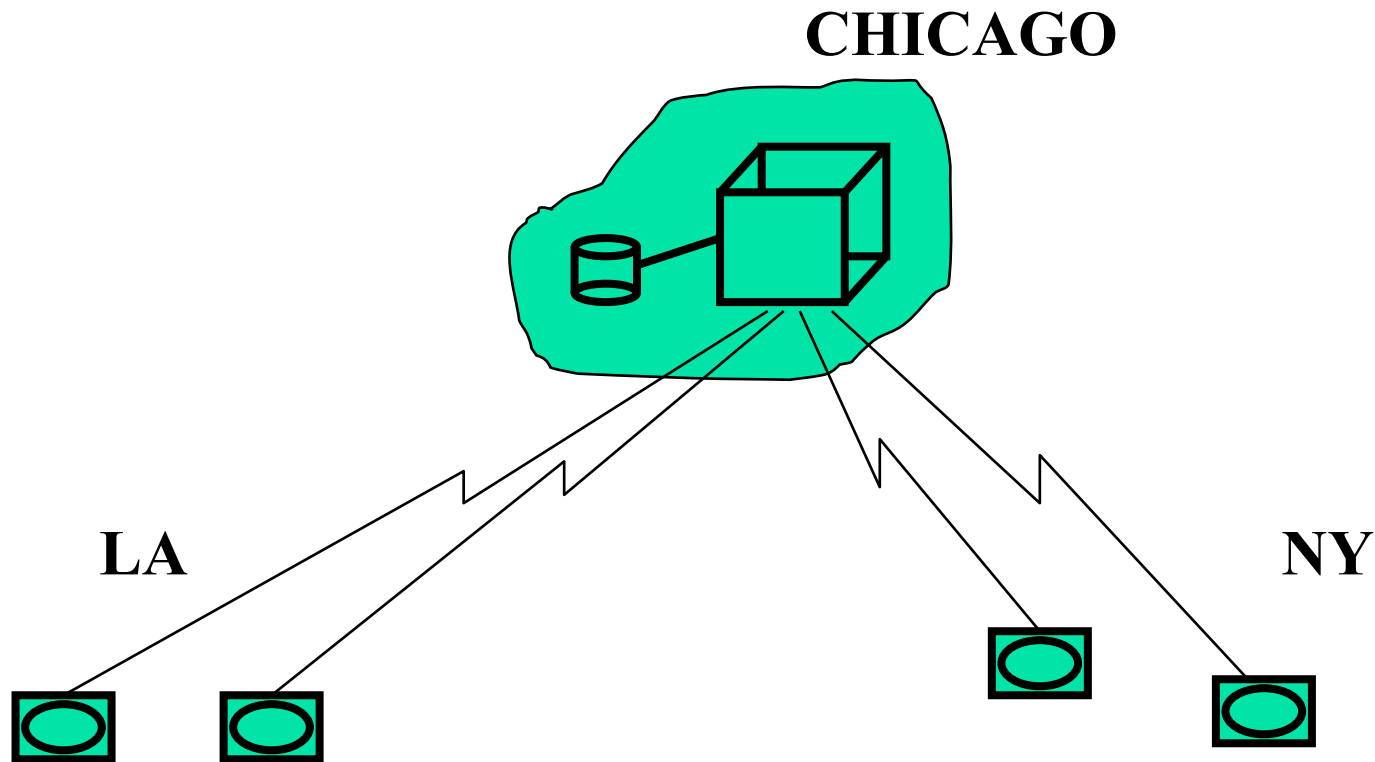
# Γενική Επισκόπηση

---

- **Πρόβλημα – κίνητρο**
- Θέματα σχεδιασμού
- Επεξεργασία ερωτήματος – Ημιενώσεις
- Δοσοληψίες (ανάκαμψη, έλεγχος συνδρομικότητας)

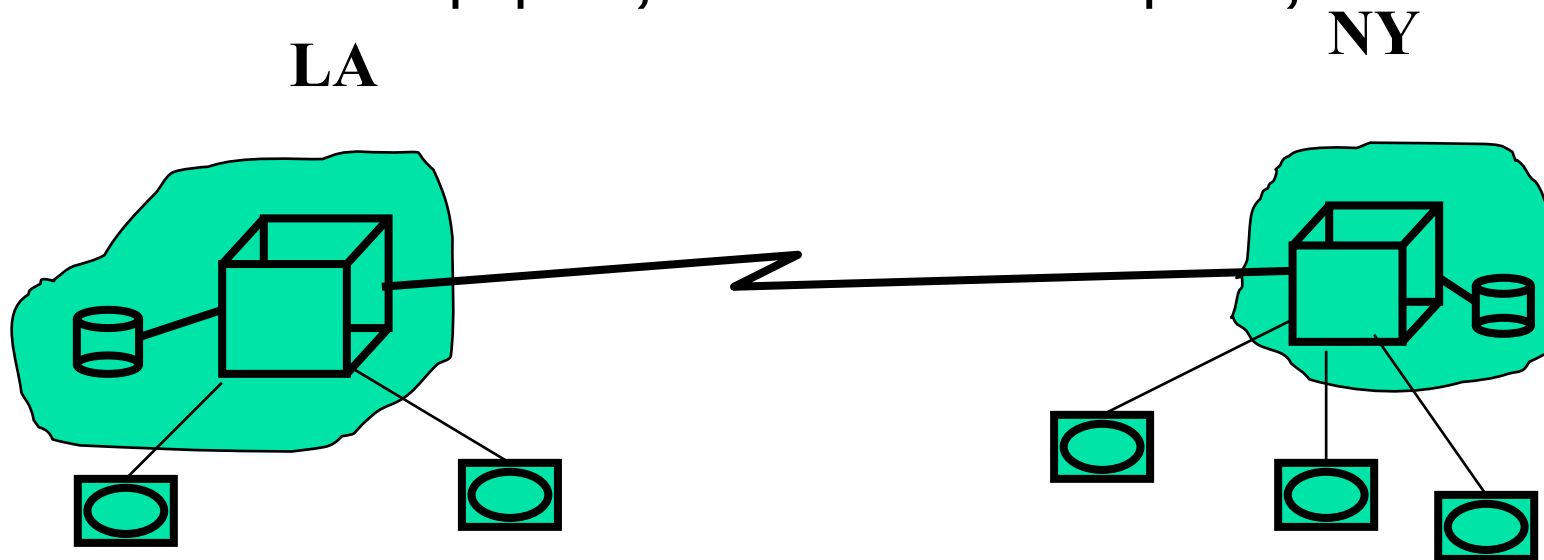
# Πρόβλημα – ορισμός

- Κεντριοποιημένη ΒΔ (centralized DB):



# Πρόβλημα – ορισμός

- Κατανεμημένη ΒΔ (distributed DB):
  - Τα δεδομένα (ΒΔ) αποθηκεύονται σε διαφορετικούς τόπους (sites)
  - ... οι επιμέρους ΒΔ είναι συνδεδεμένες



# Πρόβλημα – ορισμός



## ■ Κατανεμημένη ΒΔ:

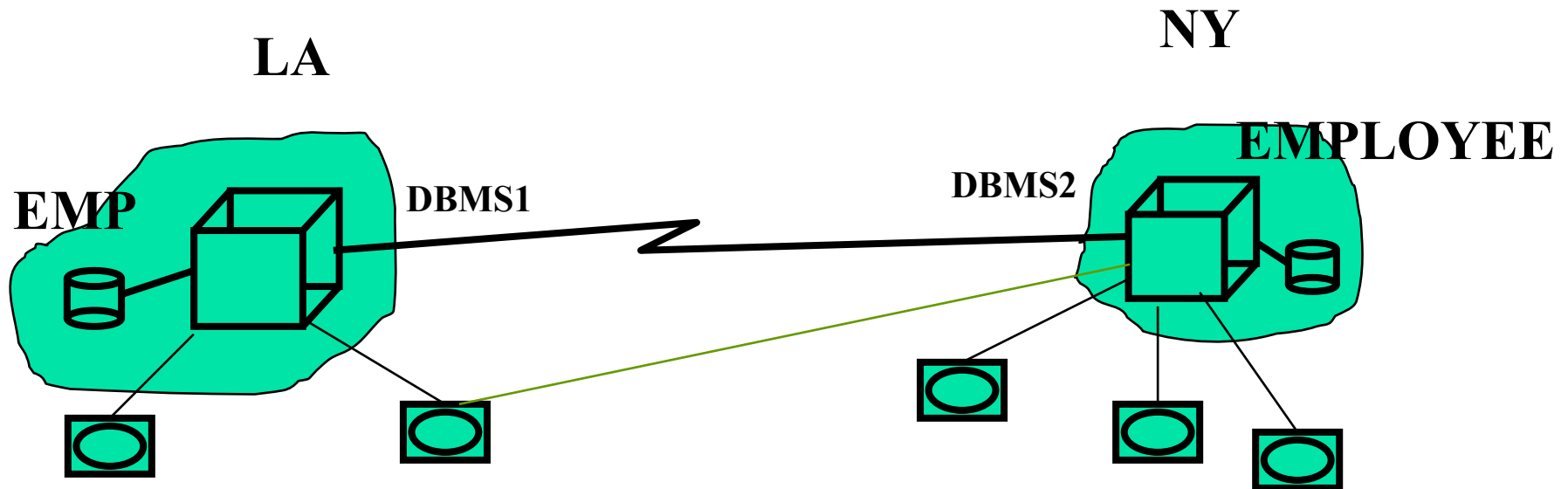
- Αποθηκεύεται σε πολλές συνδεδεμένες ΒΔ
- Τυπικά γεωγραφικά διαχωρισμένες
- Με ξεχωριστή διαχείριση
- Οι δοσοληψίες διακρίνονται σε:
  - Τοπικές (Local)
  - Γενικές (Global)
- Πιθανώς διαφορετικά ΣΔΒΔ, σχήματα ΒΔ (ετερογενή)

# Πρόβλημα – ορισμός

τώρα:

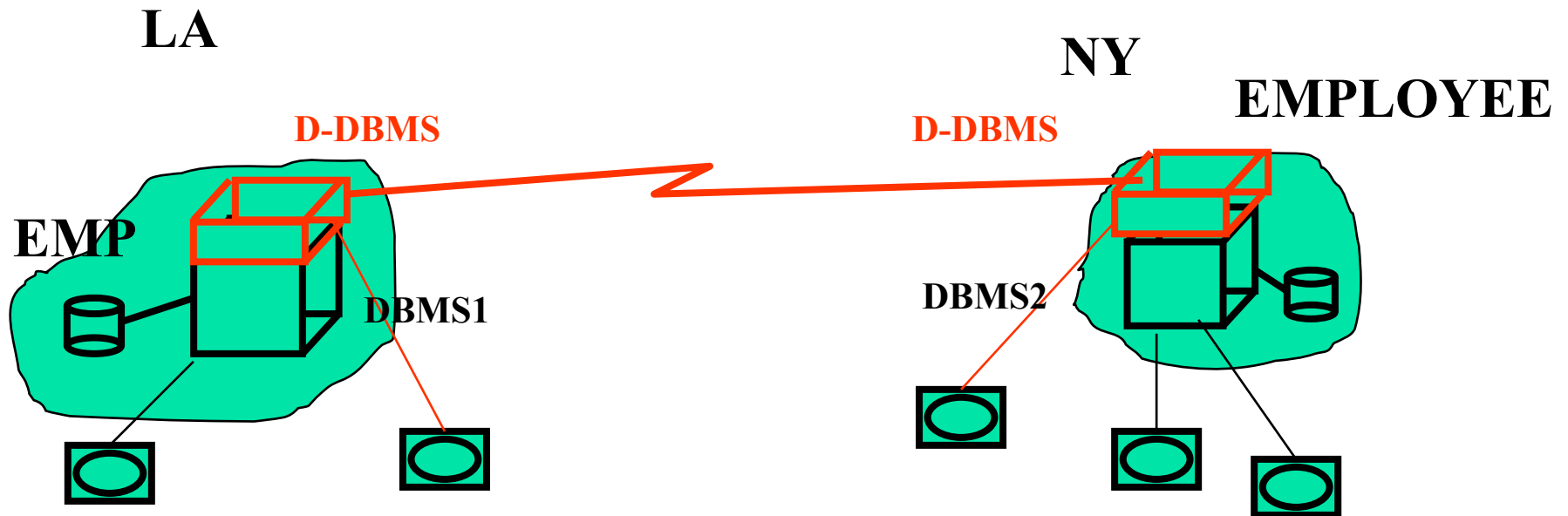
```
connect to LA; exec sql select * from EMP; ...
```

```
connect to NY; exec sql select * from EMPLOYEE; ...
```



# Πρόβλημα- ορισμός

ιδανικά: `connect to distr-LA; exec sql select * from EMPL;`







# Πλεονεκτήματα και μειονεκτήματα ;

---



# Πλεονεκτήματα + Μειονεκτήματα

---

## ■ Πλεονεκτήματα

- Διαμοίραση δεδομένων
- Αξιοπιστία & διαθεσιμότητα
- Αυτονομία (τοπική)
- Πιο γρήγορη επεξεργασία ερωτήματος

## ■ Μειονεκτήματα

- Κόστος ανάπτυξης λογισμικού
- Αυξημένη πιθανότητα για bugs
- Επιπλέον επεξεργασία (μηνύματα)/ συντήρηση



# Γενικά

---

- Πρόβλημα – κίνητρο
- **Σχεδιαστικά θέματα**
- Επεξεργασία ερωτήματος– semijoins
- Δοσοληψίες (ανάκαμψη, έλεγχος συνδρομικότητας)



# Σχεδιασμός κατακεμημένου ΣΔΒΔ

---

- Ομογενείς κατακεμημένες ΒΔ
  - Ίδια ΣΔΒΔ
  - Ίδιο σχήμα ΒΔ
  - Ενημερωμένο το ένα για το άλλο
  - Συμφωνούν να συνεργαστούν κατά την επεξεργασία δοσοληψιών
- Ετερογενείς κατακεμημένες ΒΔ
  - Διαφορετικά ΣΔΒΔ
  - Διαφορετικό σχήμα ΒΔ
  - Μπορεί να μη γνωρίζει το ένα το άλλο
  - Μπορεί να παρέχουν συνεργασία κατά την επεξεργασία των δοσοληψιών



# Σχεδιασμός κατανεμημένων ΣΔΒΔ

---

Ποιες είναι οι επιλογές για την αποθήκευση ενός πίνακα?



## Σχεδιασμός κατακεμημένων ΣΔΒΔ

---

- Αντίγραφα-(ομοιοτυπα αντίγραφα ενός πίνακα σε διαφορετικά sites)
- Τεμαχισμός (οριζόντιος, κάθετος, υβριδικός)
- ή και τα δύο...



# Σχεδιασμός κατανεμημένων ΣΔΒΔ

---

- Αντίγραφα: ένα αντίγραφο μιας σχέσης αποθηκεύεται σε δύο ή περισσότερους τόπους
- Πλεονεκτήματα και μειονεκτήματα
  - Διαθεσιμότητα
  - Αυξημένος παραλληλισμός (πιθανή ελαχιστοποίηση της μετακίνησης δεδομένων μεταξύ τόπων)
  - Αυξημένο overhead κατά την ενημέρωση (τα αντίγραφα θα πρέπει να είναι συνεπή)



# Σχεδιασμός κατακεμημένων ΣΔΒΔ

Τεμαχισμός:

Κάθετος  
τεμαχισμός

ssn	Όνομα	Διεύθυνση
123	Γιάννης	Μαιζώνος
...	...	...
234	Γιώργος	Κορίνθου

Οριζόντιος  
τεμαχισμός

- διατήρηση των πλειάδων/γνωρισμάτων στους τόπους που χρησιμοποιούνται περισσότερο
- επιβεβαίωση ότι ο πίνακας μπορεί να ξανακατασκευαστεί





# Διαφάνεια & Αυτονομία

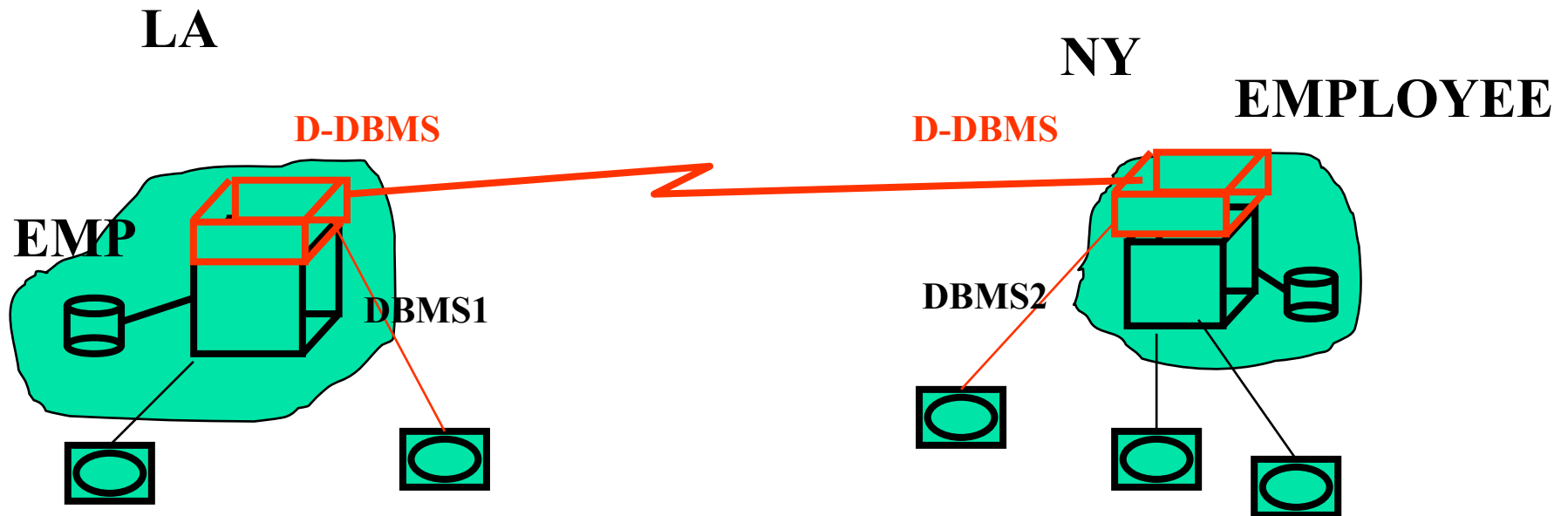
---

Θέματα/Στόχοι:

- Ονοματισμός και τοπική αυτονομία
- Διαφάνεια αντιγραφής
- Διαφάνεια τεμαχισμού
- Διαφάνεια τόπου

# Πρόβλημα- Ορισμός

ιδανικά: `connect to distr-LA; exec sql select * from EMPL;`





# Γενική Επισκόπηση

---

- Πρόβλημα – κίνητρο
- Θέματα σχεδιασμού
- **Επεξεργασία ερωτήματος – Ημιενώσεις**
- Δοσοληψίες (ανάκαμψη, έλεγχος συνδρομικότητας)



# Κατανεμημένη επεξεργασία ερωτήματος

---

- θέματα (επιπλέον αυτών για κεντριοποιημένη q-opt)

- Κόστος μετάδοσης

- (ΚΜΕ, δίσκος, #bytes που μεταδόθηκαν,

- #μηνύματα που μεταδόθηκαν)

- παραλληλισμός / επικάλυψη καθυστερήσεων

- Μείωση παρερχόμενου χρόνου;

- ή ελαχιστοποίηση της κατανάλωσης πόρων;

# Κατανεμημένη επεξεργασία ερωτήματος

S1

**SUPPLIER**

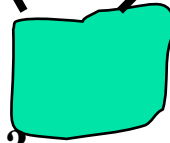
s#	...
s1	
s2	
s5	
s11	

S2

**SHIPMENT**

s#	p#
s1	p1
s2	p1
s3	p5
s2	p9

S3



**SUPPLIER Join SHIPMENT = ?**



# Ημιενώσεις (Semijoins)

---

- Επιλογή πλάνων?
- πλάνο #1: ship SHIP -> S1; join; ship -> S3
- πλάνο #2: ship SHIP->S3; ship SUP->S3; join
- ...
- Άλλα πλάνα;

# Κατανεμημένη βελτίωση ερωτήματος – semijoins

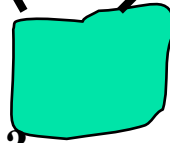
S1

SUPPLIER	
s#	...
s1	
s2	
s5	
s11	

S2

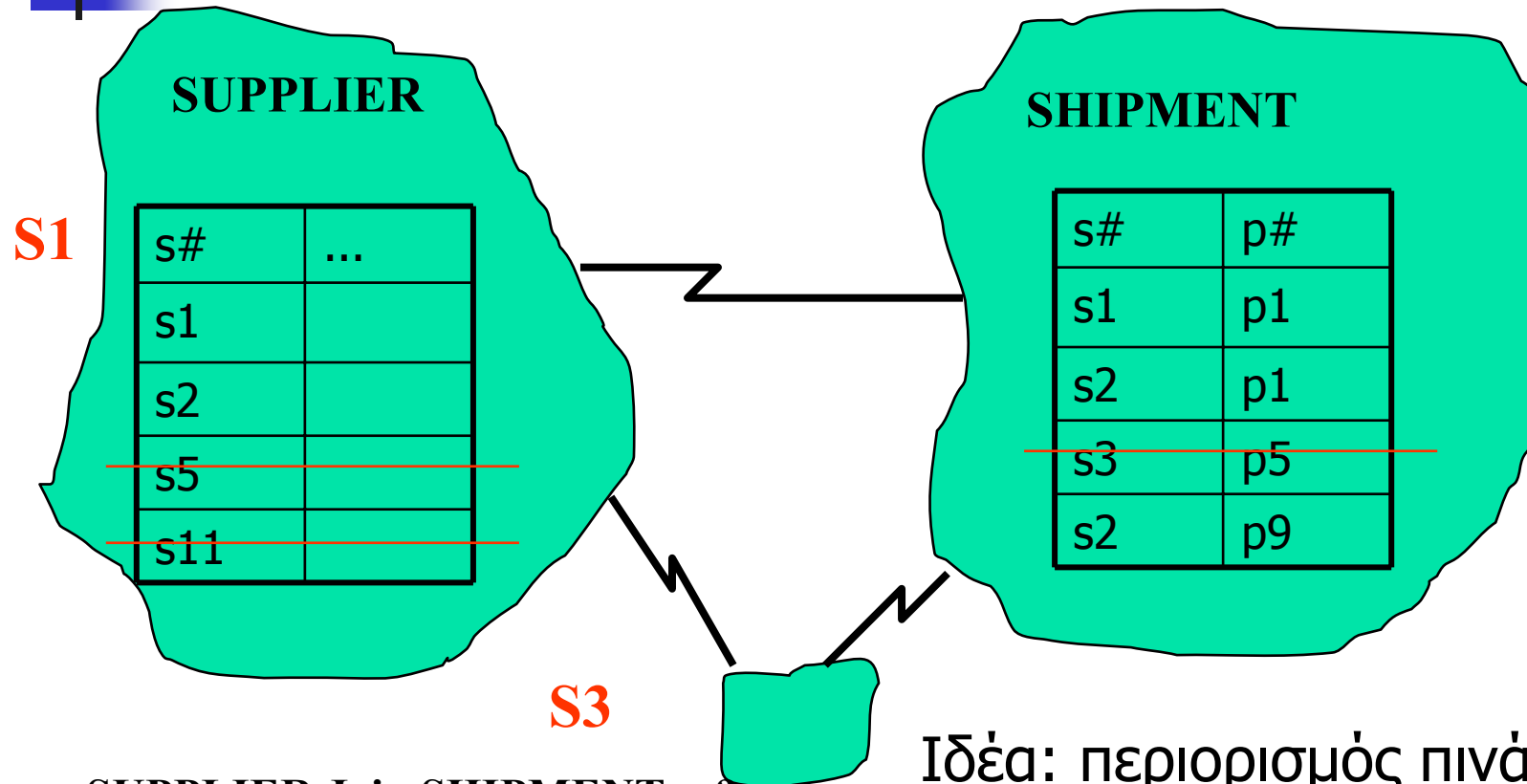
SHIPMENT	
s#	p#
s1	p1
s2	p1
s3	p5
s2	p9

S3



SUPPLIER Join SHIPMENT = ?

# Ημιενώσεις



Ιδέα: περιορισμός πινάκων πριν τη μεταφορά



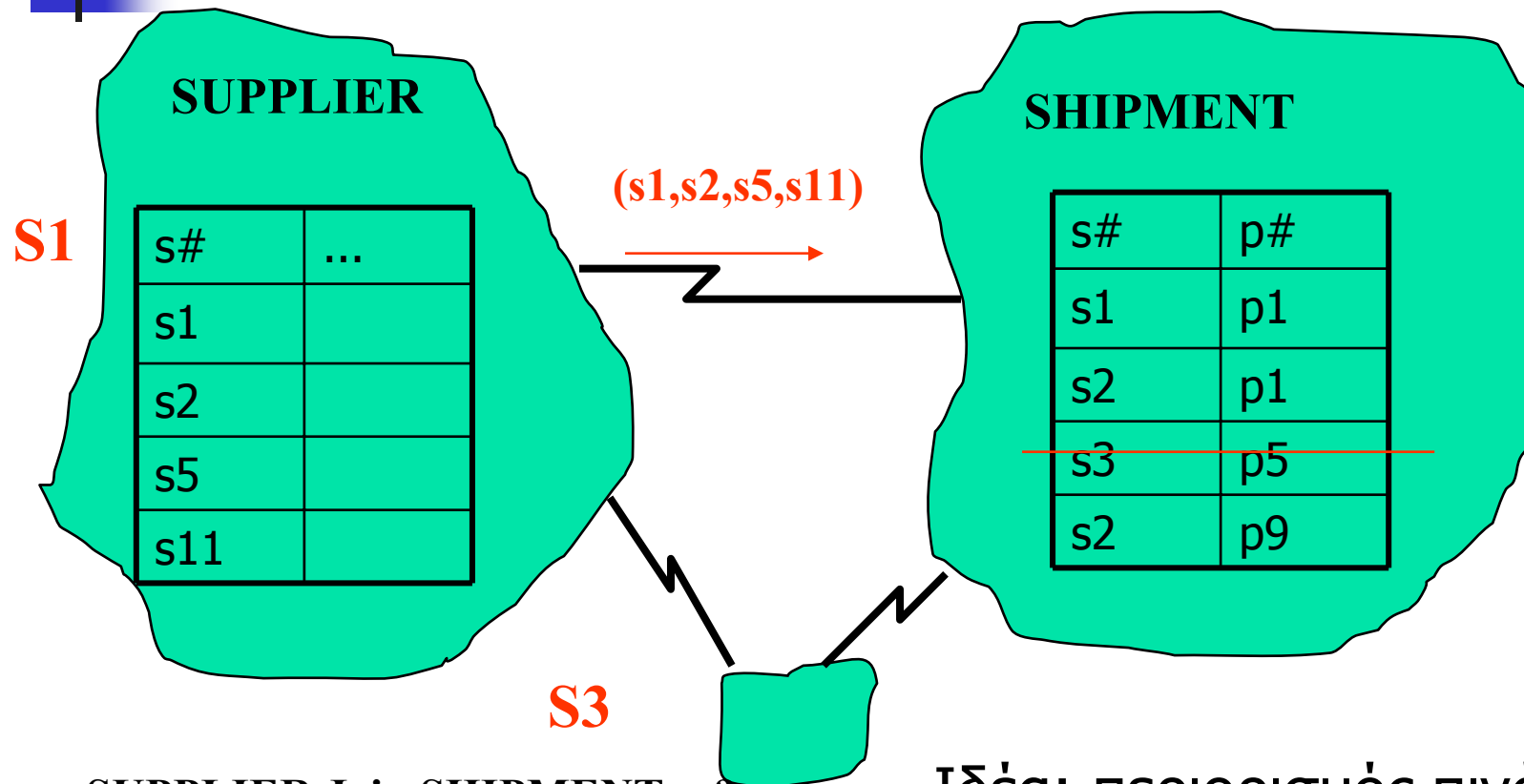


# Ημιενώσεις

---

- Πώς γίνεται η περικοπή, με μικρό κόστος?
- Π.χ., περιορισμός του πίνακα 'SHIPMENT':

# Ημιενώσεις



SUPPLIER Join SHIPMENT = ?

Ιδέα: περιορισμός πινάκων πριν τη μεταφορά



# Ημιενώσεις

---

- Τυπικά:
- $SHIPMENT' = SHIPMENT \triangleright \triangleleft SUPPLIER$
- Έκφραση ημιένωσης με σχεσιακή άλγεβρα



# Ημιενώσεις

---

- Τυπικά:
- $SHIPMENT' = SHIPMENT \triangleright \triangleleft SUPPLIER$
- Έκφραση ημιένωσης με σχεσιακή άλγεβρα

$$R' = R \triangleright \triangleleft S$$

$$= \pi_R (R \triangleright \triangleleft S)$$

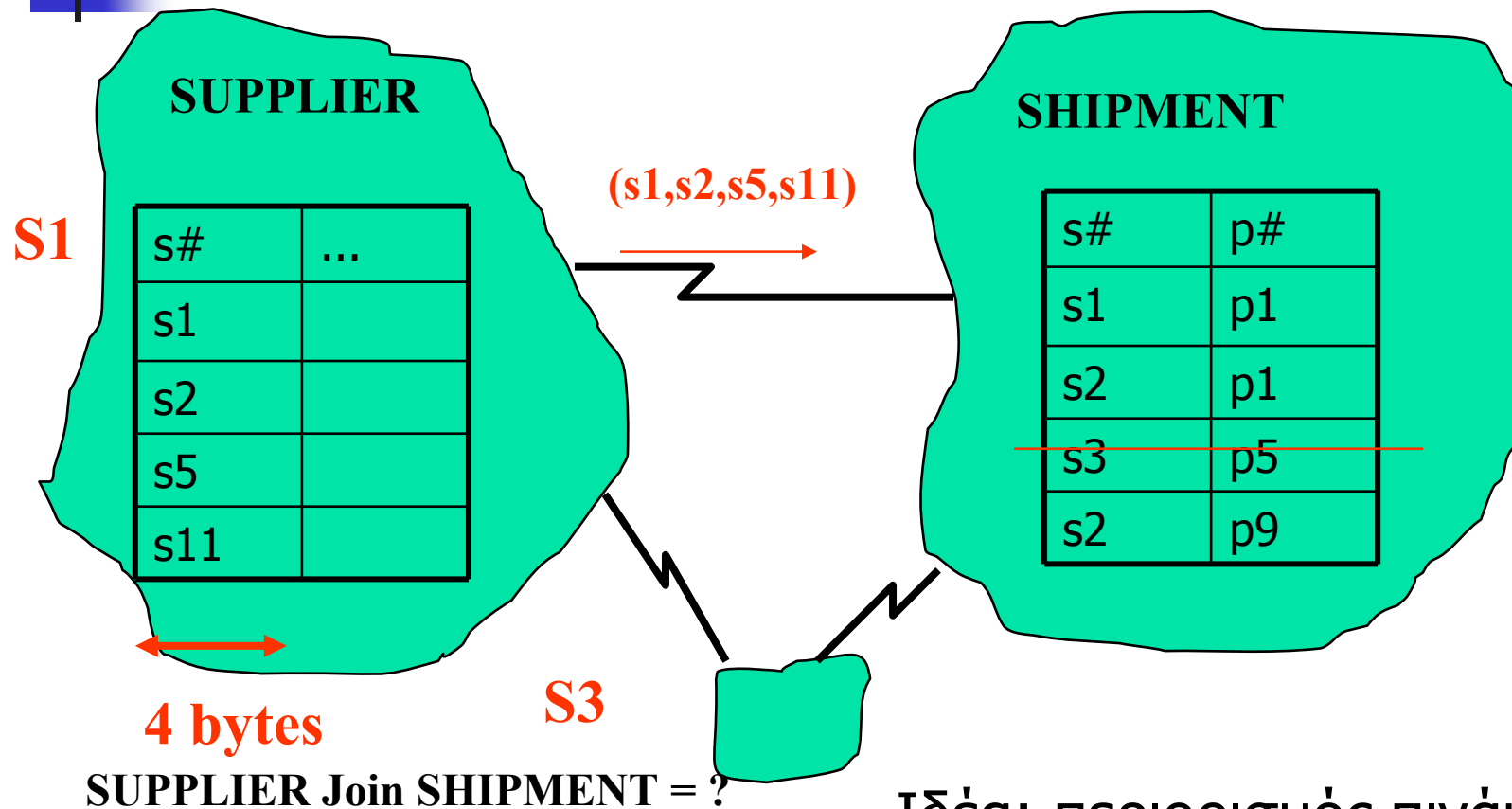


## Ημιενώσεις– π.χ.:

---

- Έστω ότι κάθε ιδιότητα είναι 4 bytes
- Q: κόστος μετάδοσης (#bytes) για ημιένωση  
SHIPMENT' = SHIPMENT semijoin SUPPLIER

# Ημιενώσεις



Ιδέα: περιορισμός πινάκων πριν τη μεταφορά



## Ημιενώσεις– π.χ.:

---

- Έστω ότι κάθε γνώρισμα είναι 4 bytes
- Q: κόστος μετάδοσης (#bytes) για ημιένωση  
SHIPMENT' = SHIPMENT semijoin SUPPLIER
- A: 4\*4 bytes



## Ημιενώσεις– π.χ.:

---

- έστω ότι κάθε γνώρισμα είναι 4 bytes
- Q1: δώσε ένα πλάνο, με ημιένωση(-εις)
- Q2: εκτίμηση του κόστους (#bytes shipped)



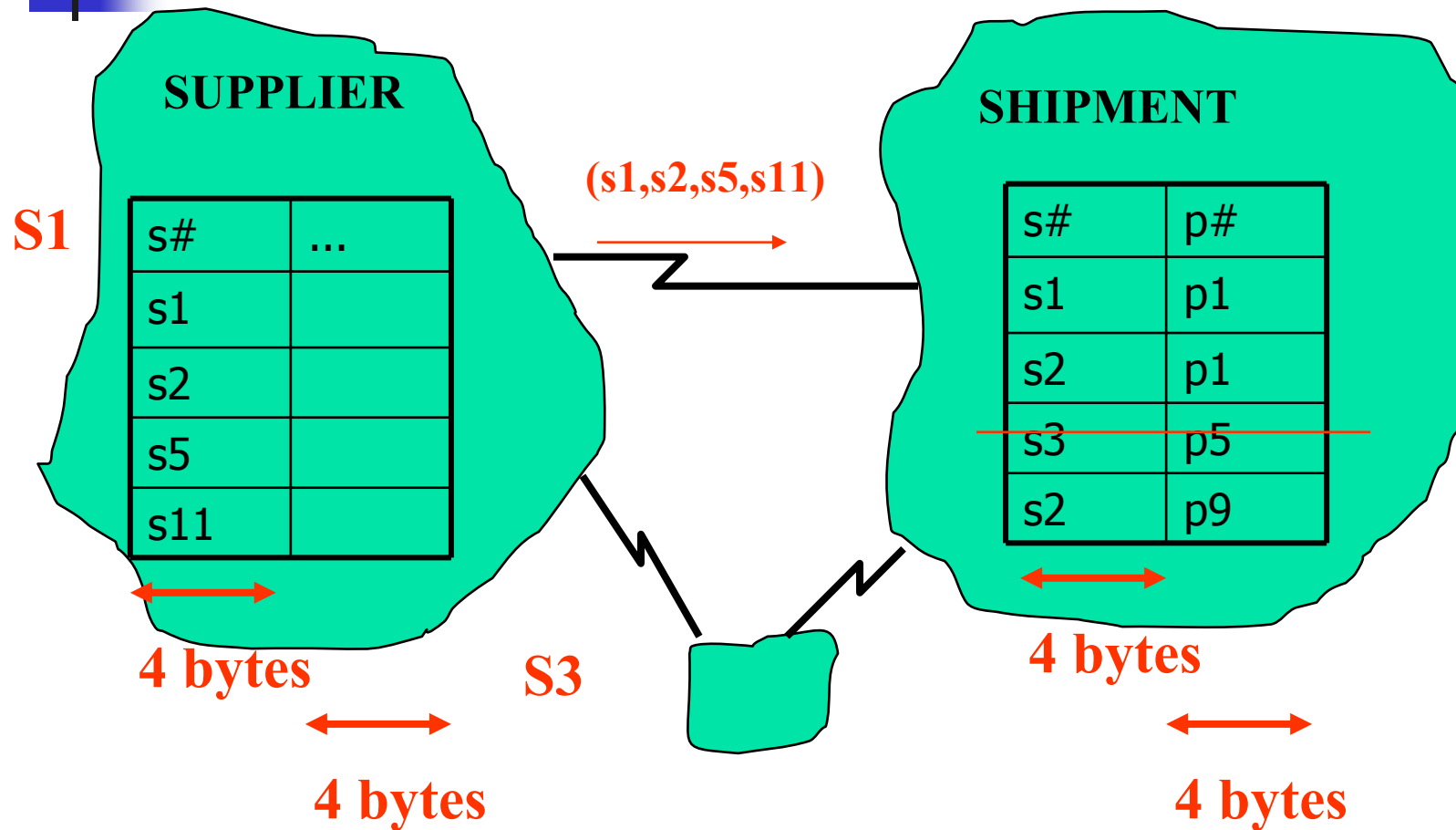


# Ημιενώσεις– π.χ.:

---

- A1:
  - Reduce SHIPMENT to SHIPMENT'
  - SHIPMENT' -> S3
  - SUPPLIER -> S3
  - do join @ S3
- Q2: κόστος;

# Ημιενώσεις





## Ημιενώσεις– π.χ.:

---

- A2:
  - 4\*4 bytes - reduce SHIPMENT to SHIPMENT'
  - 3\*8 bytes - SHIPMENT' -> S3
  - 4\*8 bytes - SUPPLIER -> S3
  - 0 bytes - do join @ S3

---

**72 bytes TOTAL**



Άλλα πλάνα;

---



## Άλλα πλάνα

---

P2:

- reduce SHIPMENT to SHIPMENT'
- reduce SUPPLIER to SUPPLIER'
- SHIPMENT' -> S3
- SUPPLIER' -> S3



## Άλλα πλάνα

---

P3:

- reduce SUPPLIER to SUPPLIER'
- SUPPLIER' -> S2
- do join @ S2
- ship results -> S3



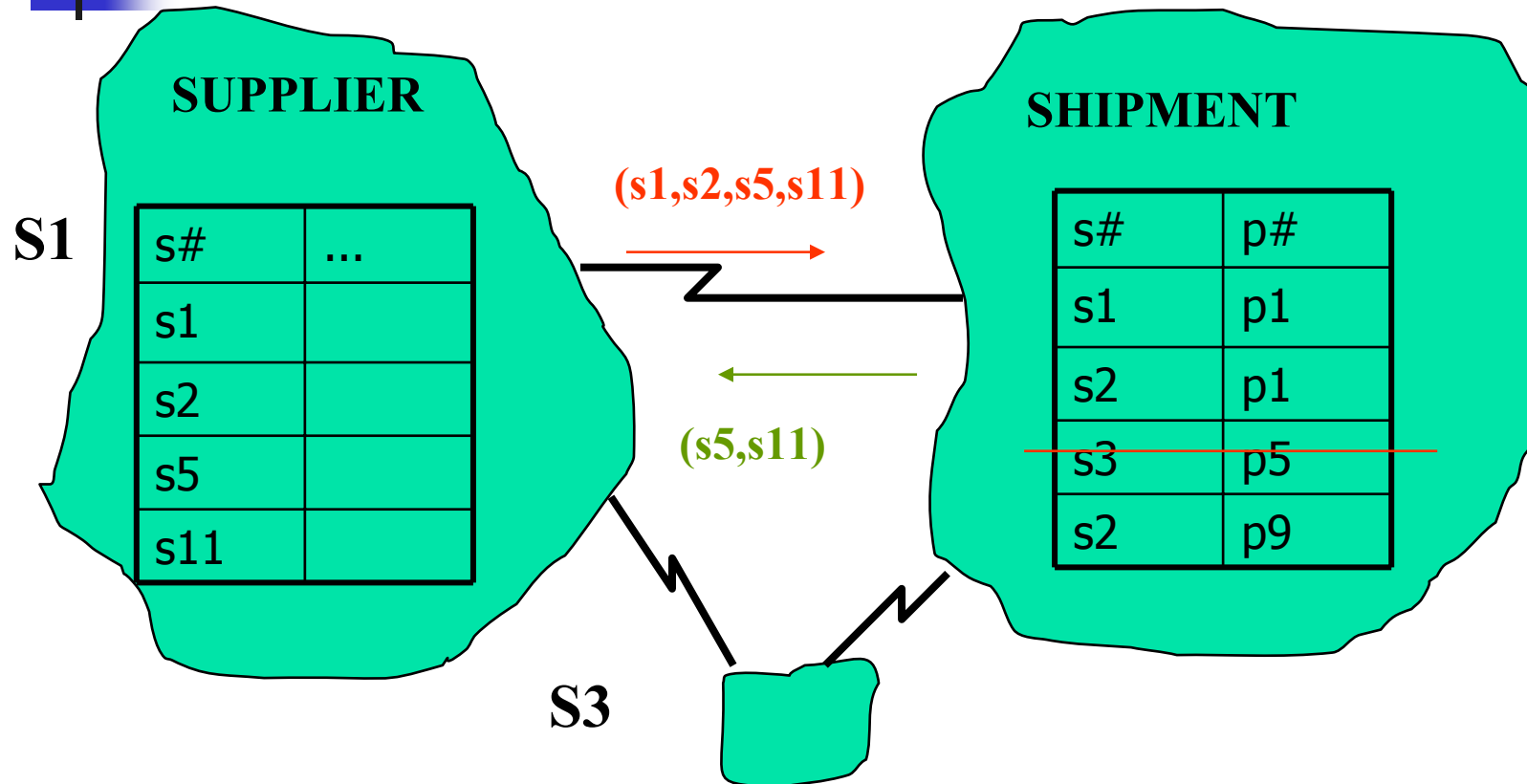
## Μία καλή ιδέα: ημιένωση δύο κατευθύνσεων

---

- Περικοπή και των δύο σχέσεων με μία ακόμα ανταλλαγή: [Kang, '86]
- Επιστροφή της λίστας κλειδιών που δεν ταίριαξε
- Δεν μπορεί να βγει χαμένη! (Γιατί?)
- Επιπλέον βελτίωση:
  - Ή της λίστας αυτών που ταίριαξαν— ό,τι είναι πιο μικρό!

# Ημιενώσεις δύο κατευθύνσεων

S2







# Επισκόπηση

---

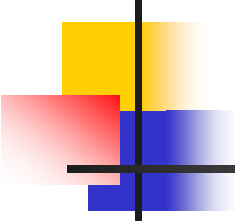
- Πρόβλημα – κίνητρο
- Σχεδιαστικά θέματα
- Επεξεργασία ερωτήματος – semi-joins
- **Δοσοληψίες (ανάκαμψη, έλεγχος συνδρομικότητας)**



# Δοσοληψίες – ανάκαμψη

---

- Πρόβλημα: π.χ., Μία δοσοληψία μετακινεί 100 € από την Αθήνα → 50 € στην Θεσσαλονίκη, 50 € στην Πάτρα
- 3 υπο-δοσοληψίες, σε 3 συστήματα
- Πώς μπορεί να εγγυηθεί η **ατομικότητα** (όλες ή καμία);
- Παρατήρηση: επιπλέον τύποι λαθών (σύνδεσμοι, εξυπηρετητές, καθυστερήσεις, διακοπές ....)

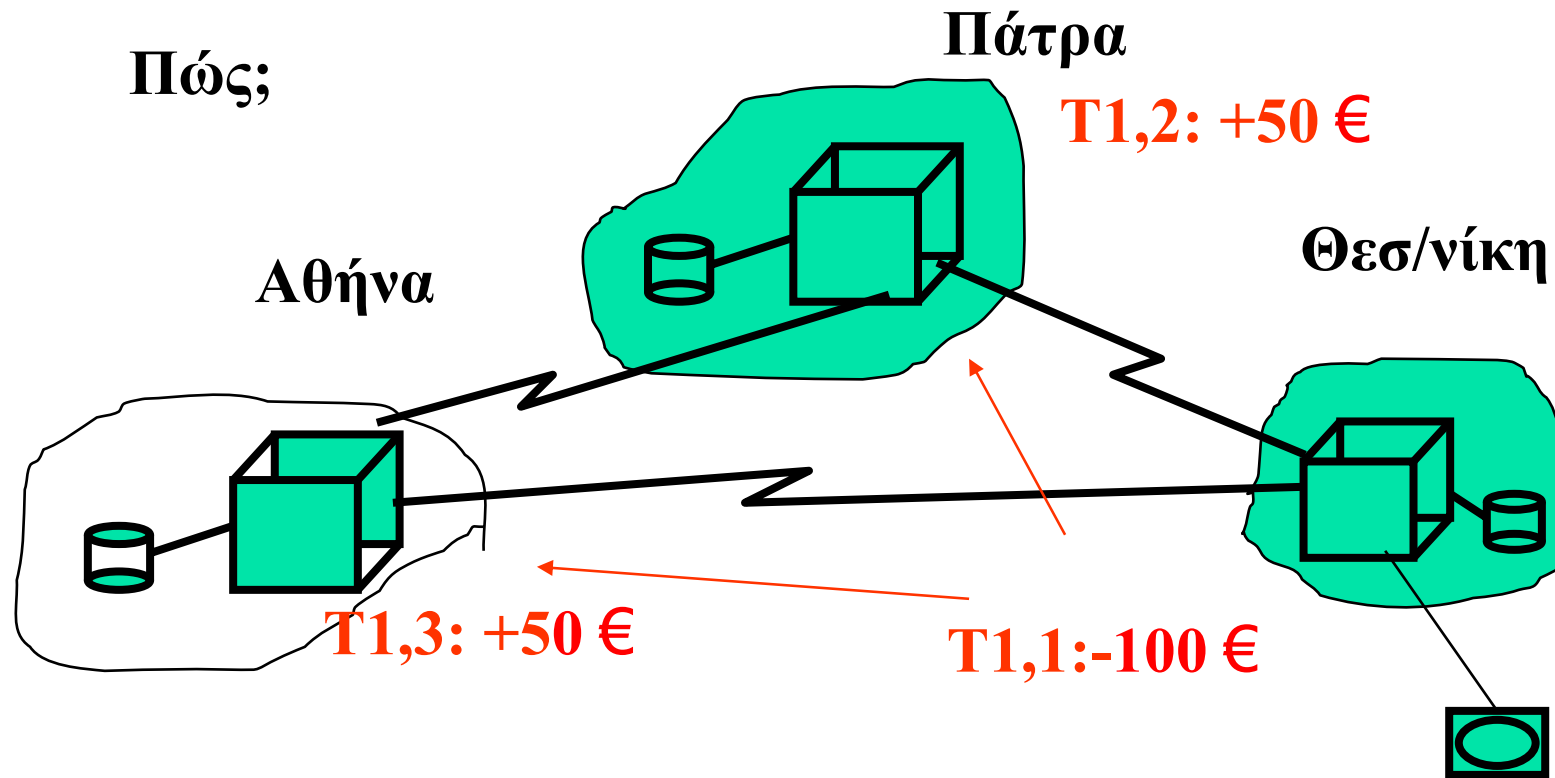


# Δοσοληψίες – ανάκαμψη

---

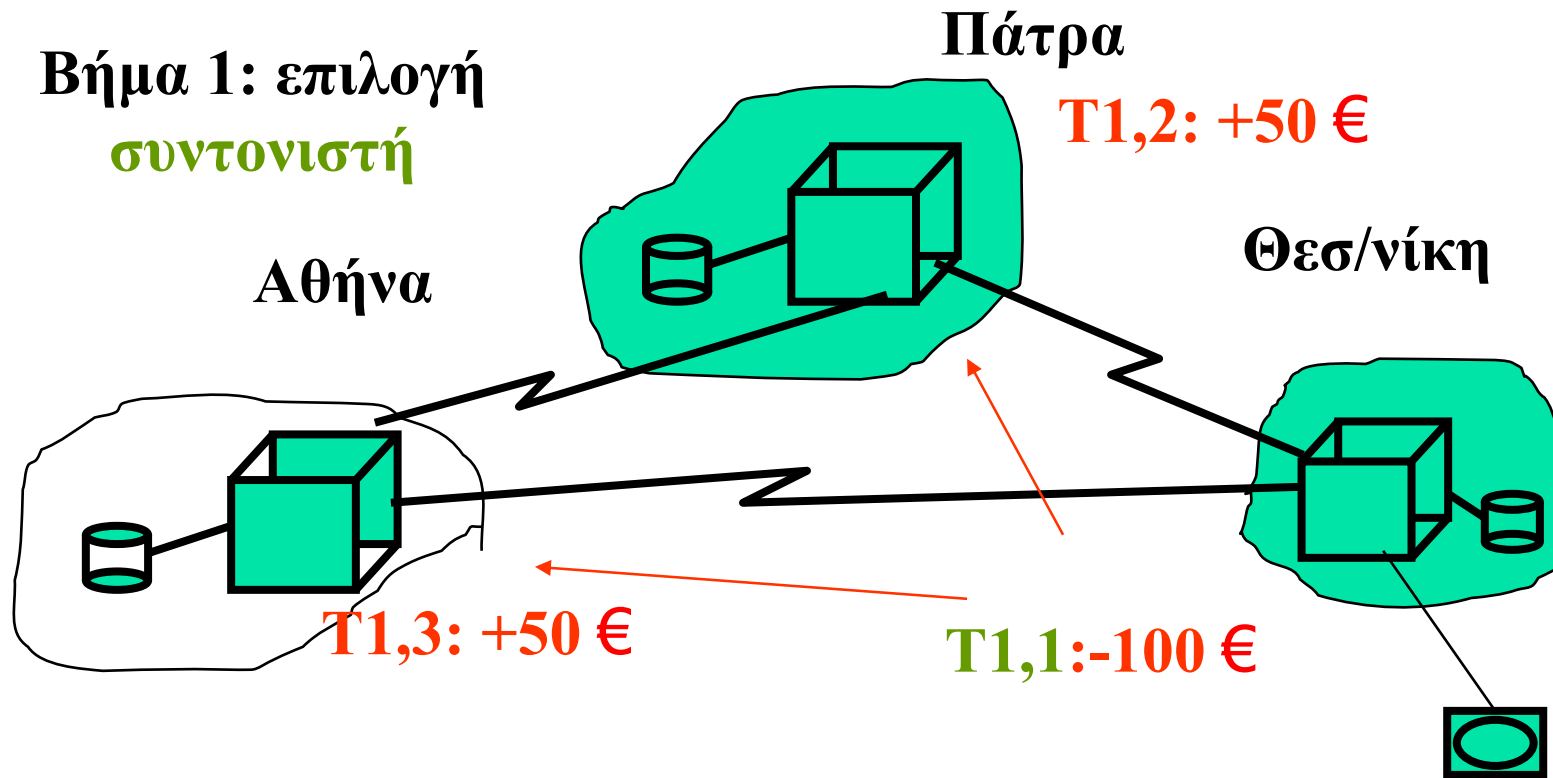
- Πρόβλημα: π.χ., μία δοσοληψία μετακινεί 100 € από την Αθήνα -> 50 € στην Θεσσαλονίκη, 50 € στην Πάτρα

# Κατανεμημένη ανάκαμψη



# Κατανεμημένη ανάκαμψη

Βήμα 1: επιλογή  
συντονιστή





# Κατανεμημένη ανάκαμψη

---

- Βήμα 2: εκτέλεση ενός πρωτόκολλου δέσμευσης,  
π.χ., “Δέσμευση δύο φάσεων” ή  
“2 Phase Commit (2PC)”  
όταν μία δοσοληψία  $T$  ολοκληρώνει την εκτέλεση (δηλαδή όταν όλες οι περιοχές στις οποίες έχει εκτελεστεί η  $T$  ενημερώνουν τον συντονιστή δοσοληψιών  $C_i$  ότι η  $T$  έχει ολοκληρωθεί)  
η  $C_i$  αρχίζει το 2PC πρωτόκολλο

# Δέσμευση 2 φάσεων

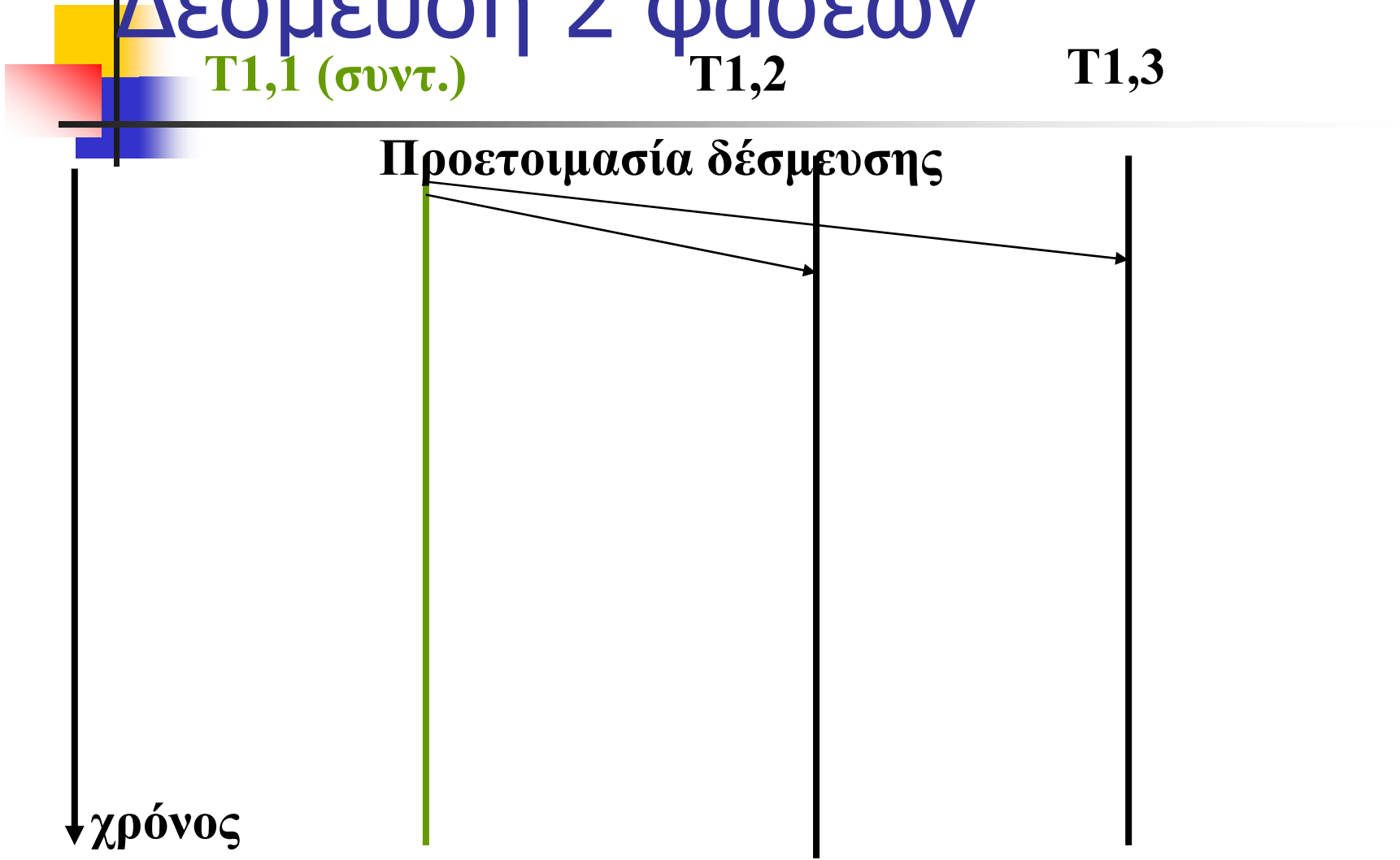
T1,1 (συντ.)

T1,2

T1,3

Προετοιμασία δέσμευσης

↓ χρόνος



# Δέσμευση 2 φάσεων

T1,1 (συντ.)

T1,2

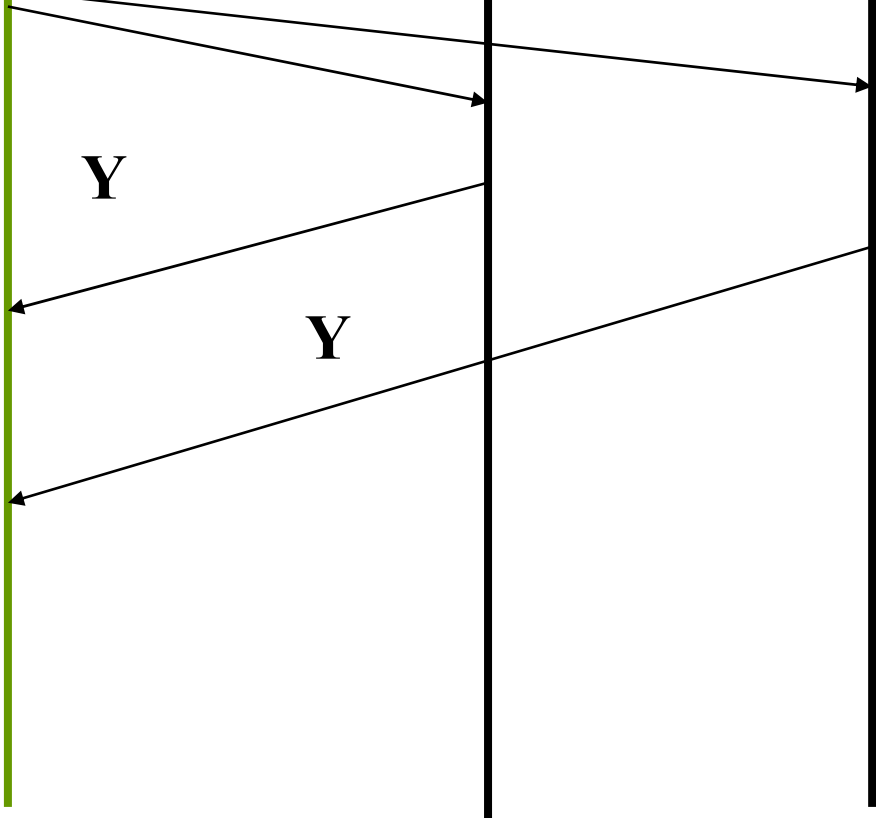
T1,3

Προετοιμασία δέσμευσης

Υ

Υ

↓ χρόνος



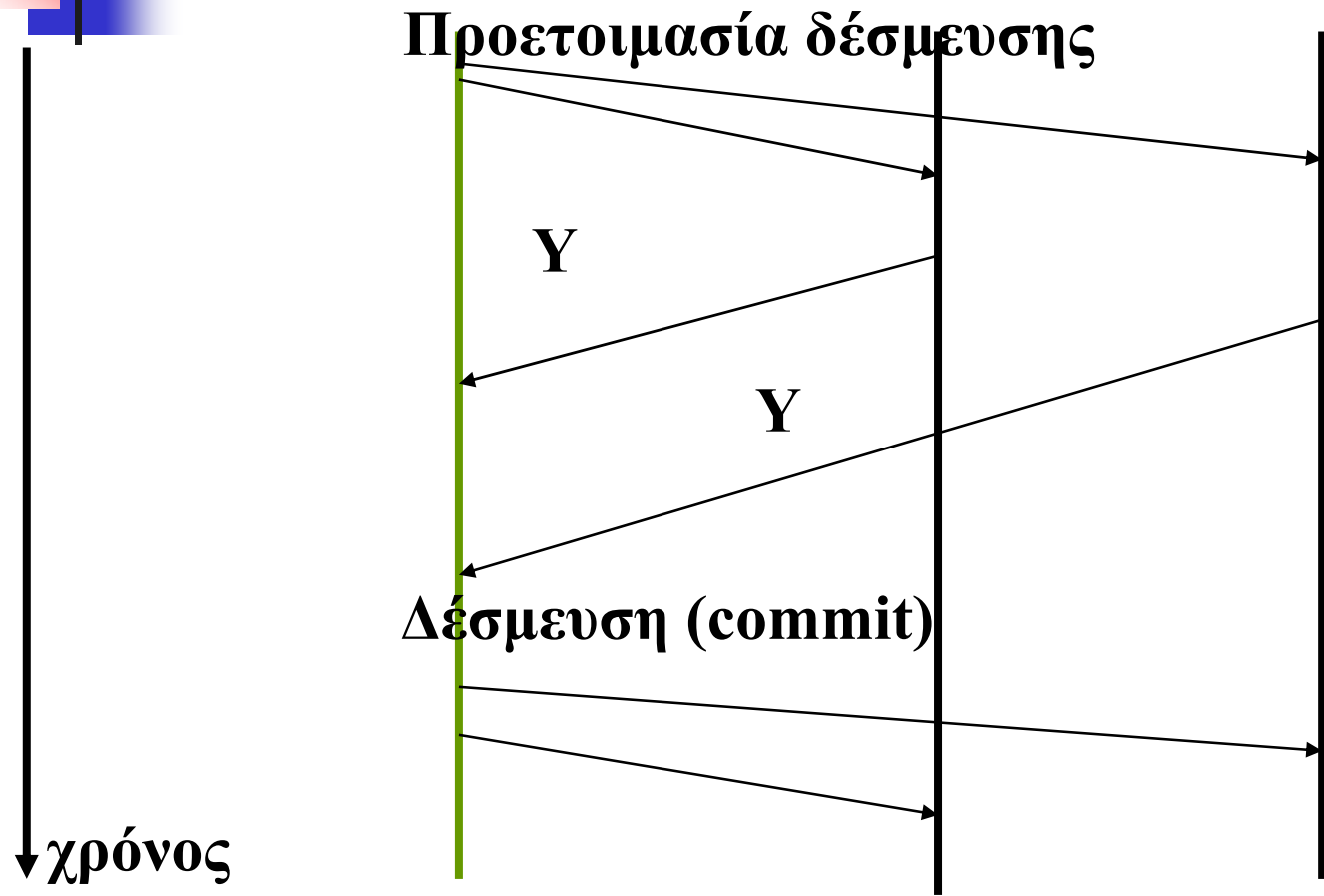


# Δέσμευση δύο φάσεων

T1,1 (συντ.)

T1,2

T1,3



# Δέσμευση 2 φάσεων (π.χ., αποτυχία)

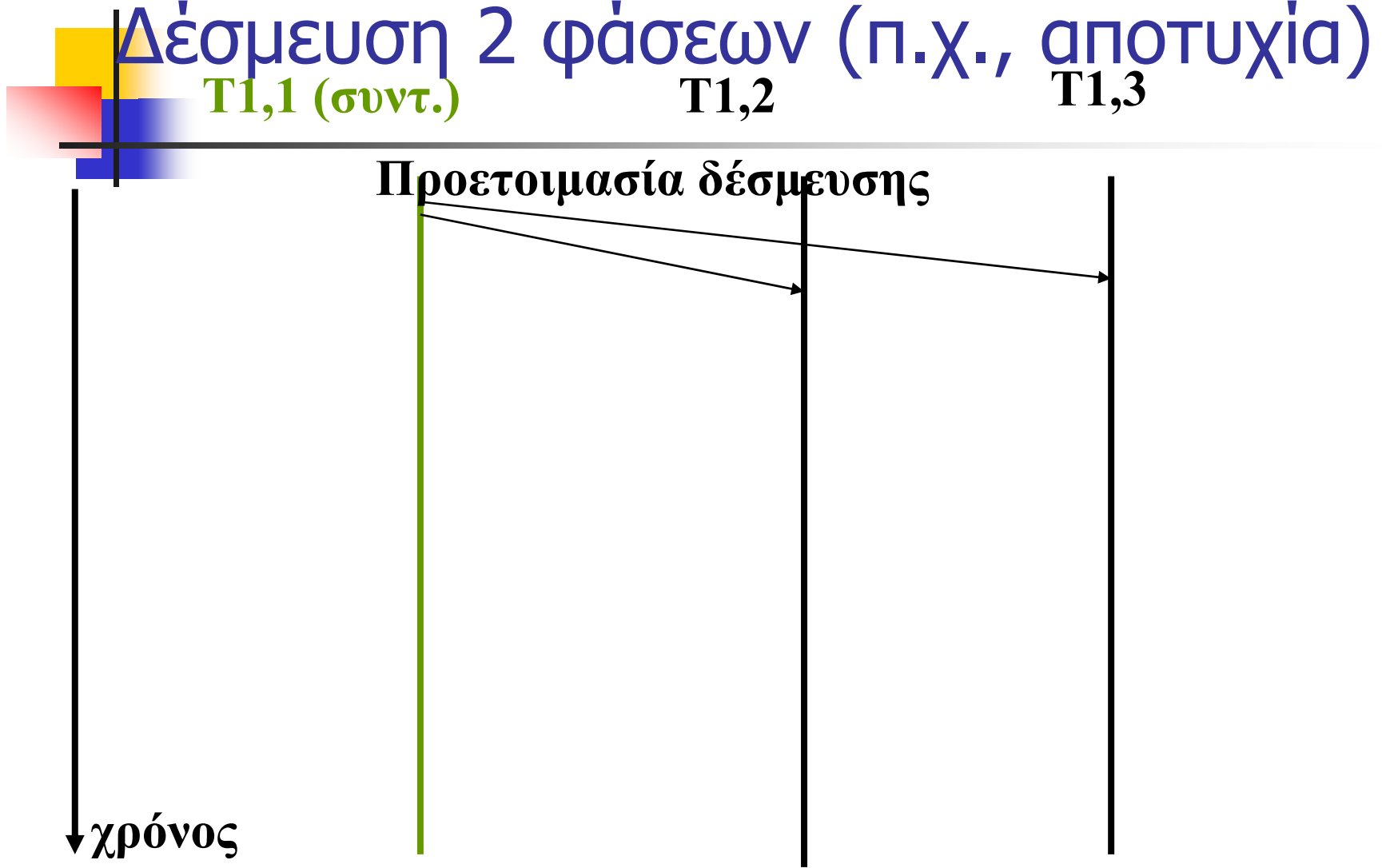
T1,1 (συντ.)

T1,2

T1,3

Προετοιμασία δέσμευσης

↓ χρόνος



# Δέσμευση 2 φάσεων

T1,1 (συντ.)

T1,2

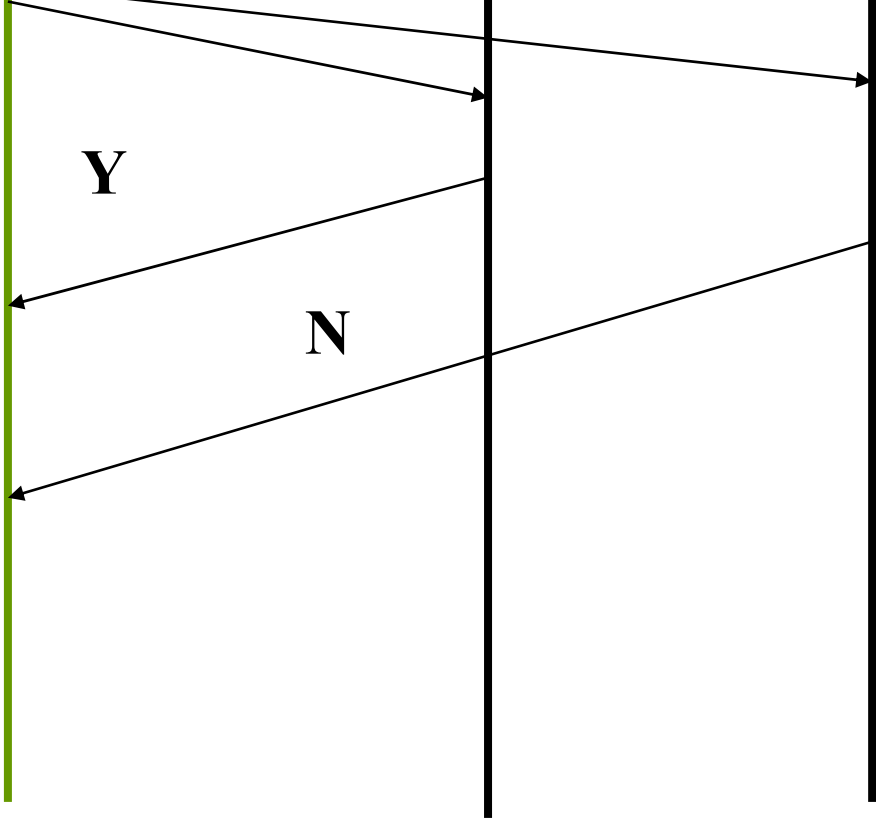
T1,3

Προετοιμασία δέσμευσης

Y

N

↓ χρόνος

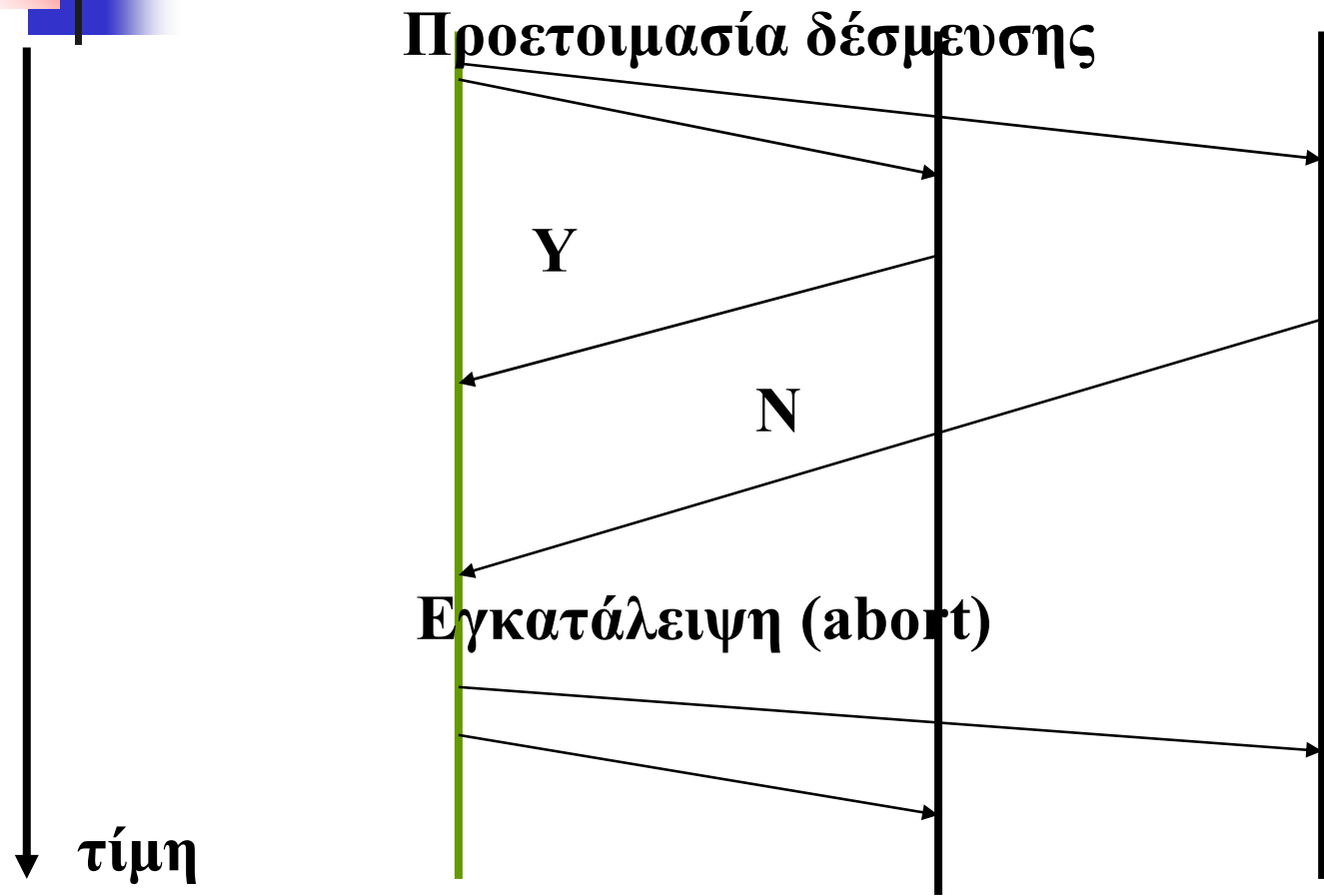


# Δέσμευση 2 φάσεων

T1,1 (coord.)

T1,2

T1,3





# Κατανεμημένη ανάκαμψη

---

- Πολλές επιπλέον λεπτομέρειες
  - τι γίνεται εάν αποτύχει ο συντονιστής;
  - τι γίνεται εάν αποτύχει η σύνδεση;
  - κτλ.
- Και πολλές άλλες λύσεις (π.χ., δέσμευση τριών φάσεων (3PC))



# Γενική επισκόπηση

---

- Πρόβλημα – κίνητρο
- Θέματα σχεδιασμού
- Επεξεργασία ερωτήματος – semi-joins
- **Δοσοληψίες** (ανάκαμψη, **έλεγχος συνδρομικότητας**)

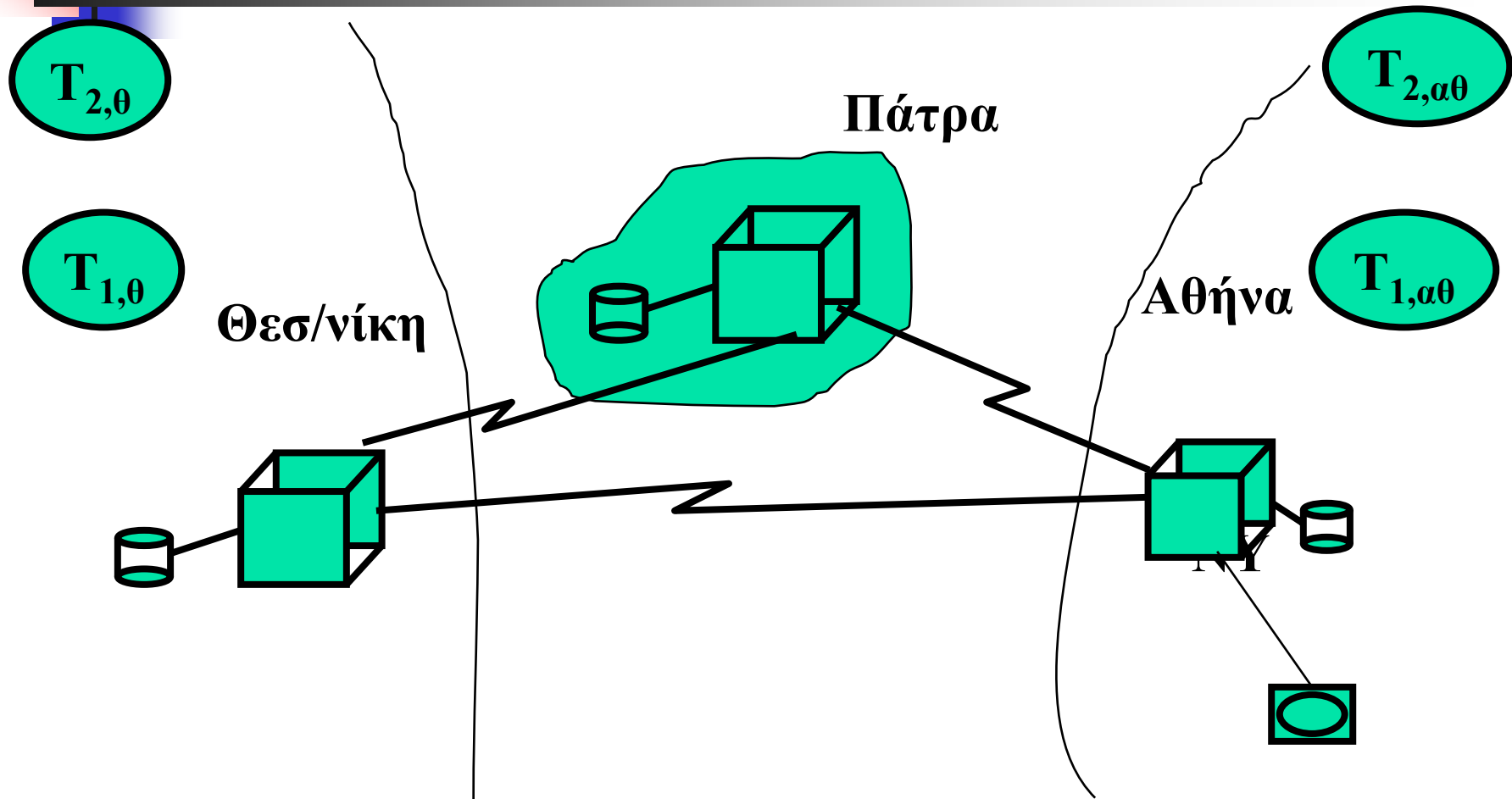


## Κατανεμημένος έλεγχος συνδρομικότητας

---

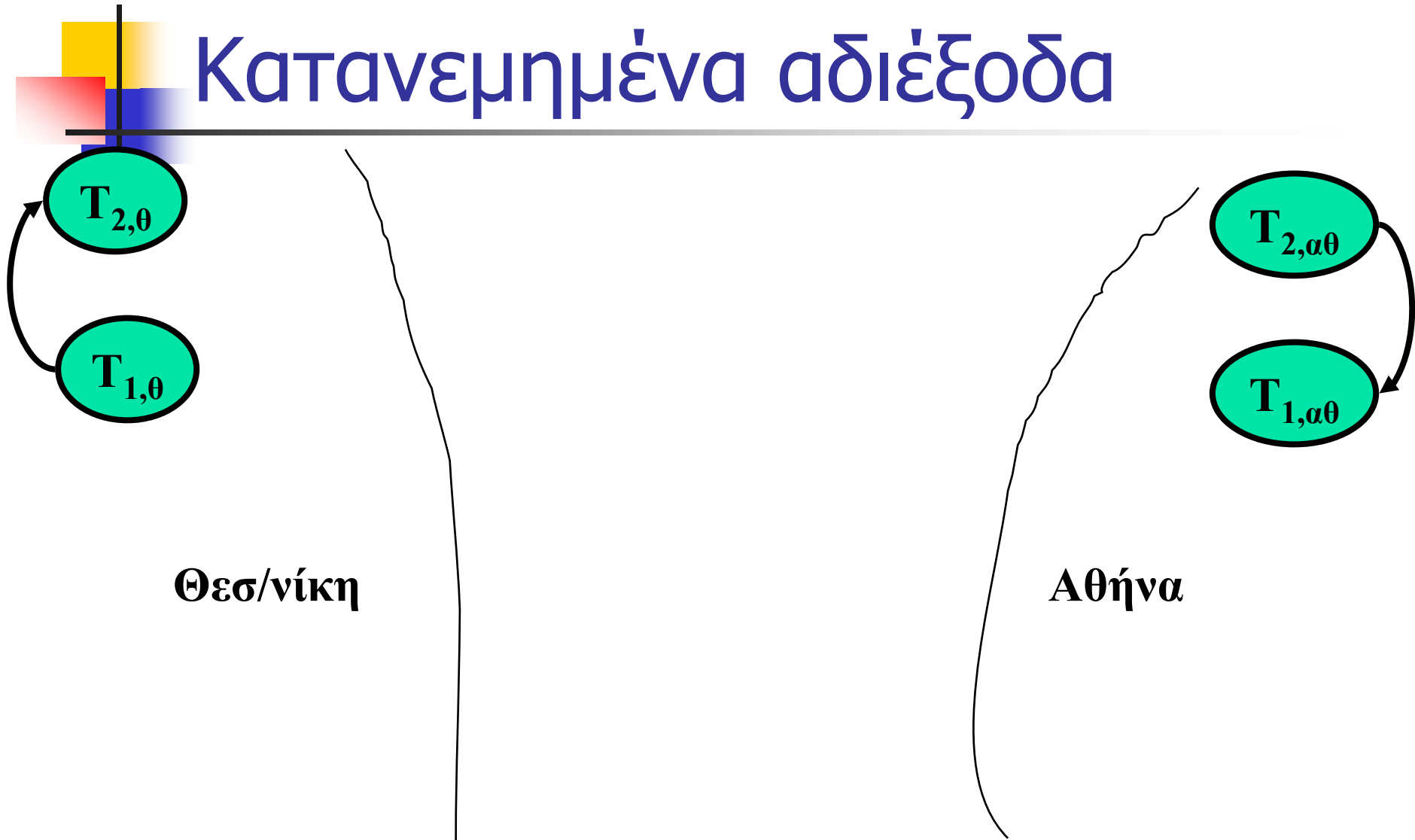
- Επίσης πιο πολύπλοκος:
- Κατανεμημένα αδιέξοδα!

# Κατανεμημένα αδιέξοδα

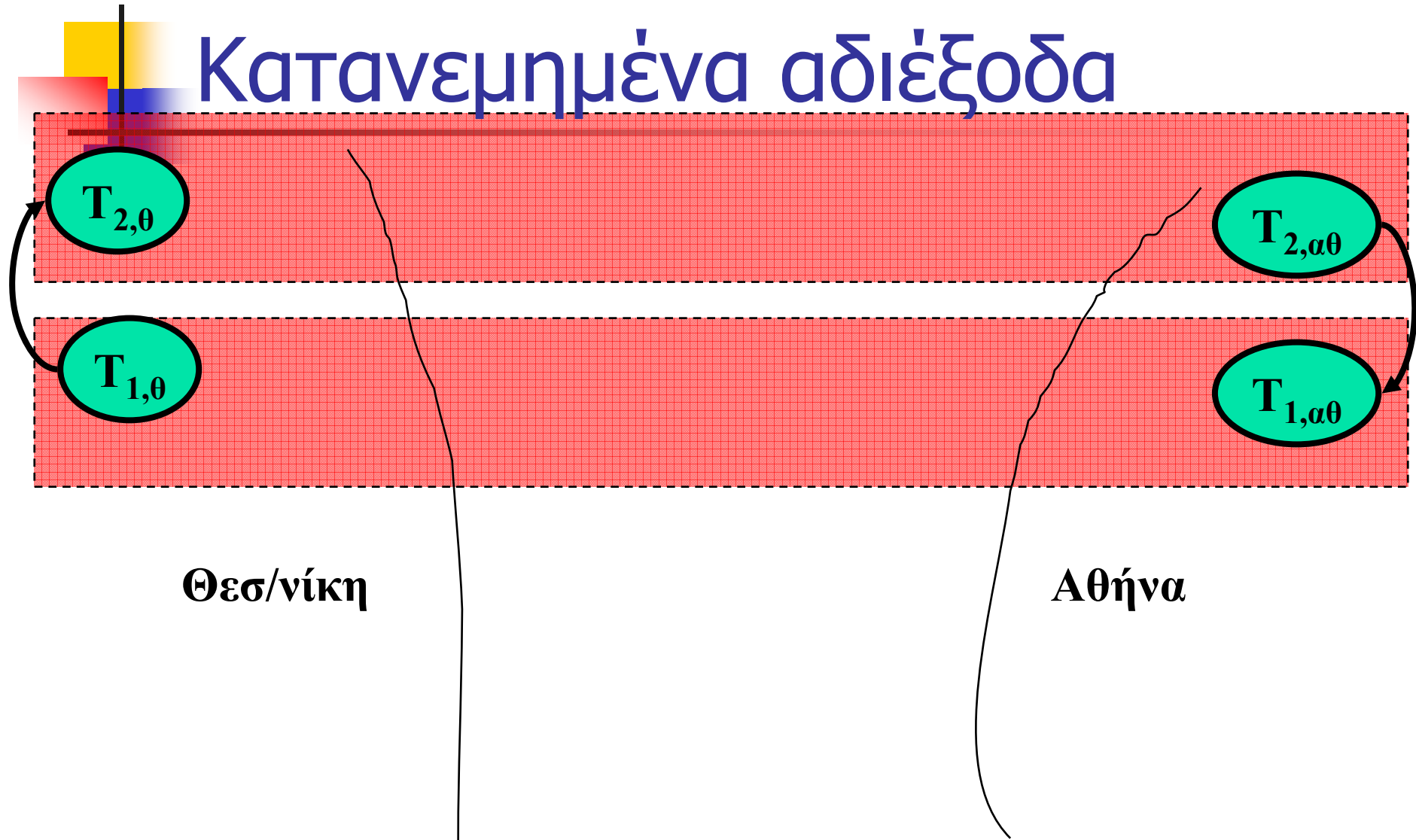




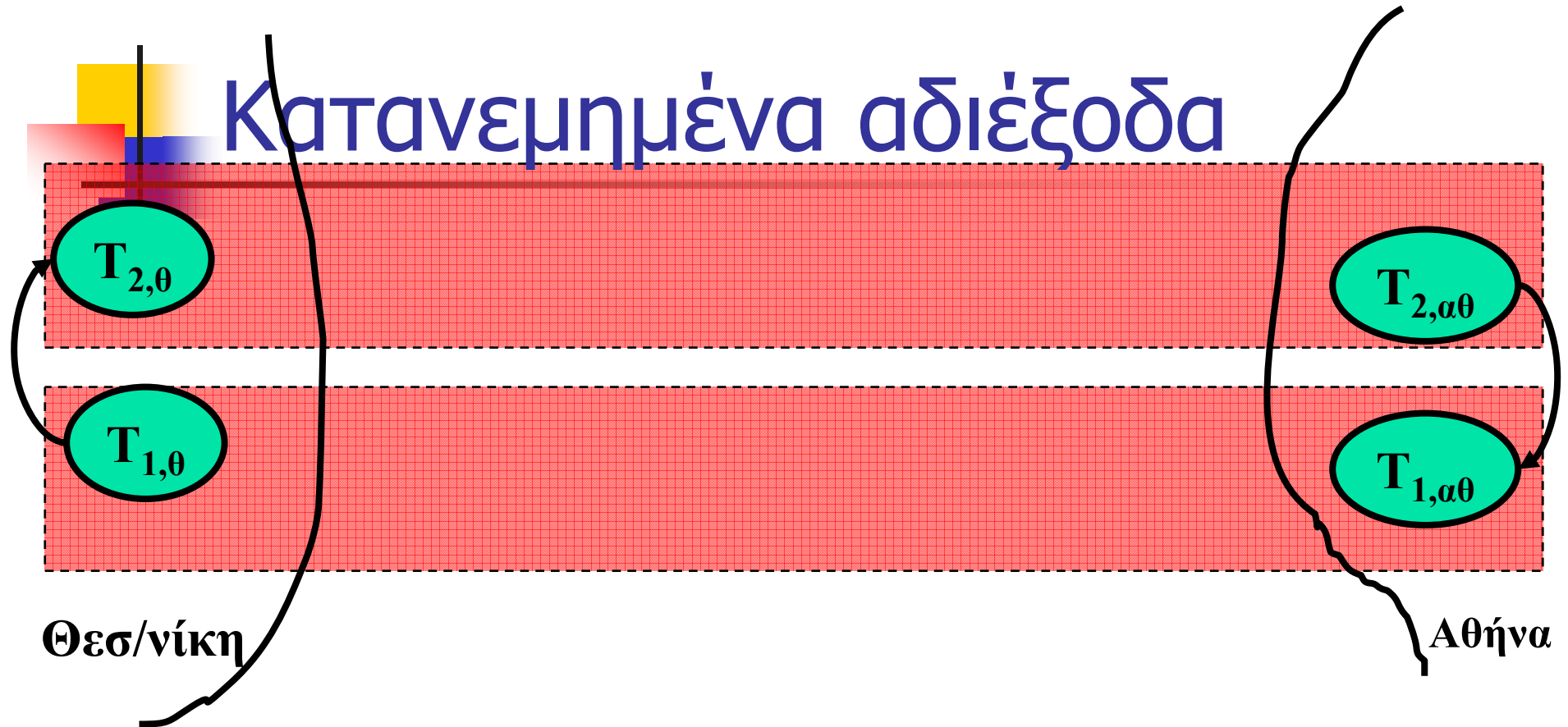
# Κατανεμημένα αδιέξοδα



# Κατανεμημένα αδιέξοδα



# Κατανεμημένα αδιέξοδα



- τα cites πρέπει να ανταλλάσσουν τα wait-for graphs τους
- έξυπνοι αλγόριθμοι, για την μείωση του # μηνυμάτων



# Συμπεράσματα

---

- Κατανεμημένα ΣΔΒΔ: not deployed
- ΑΛΛΑ: Έξυπνες ιδέες:
  - semijoins
  - Κατανεμημένη ανάκαμψη / έλεγχος συνδρομικότητας
- ..που χρησιμεύουν σε:
  - Παράλληλες ΒΔ / ομάδες-clusters ΒΔ
  - 'ενεργούς δίσκους'
  - Πανομοιότυπες ΒΔ (e-commerce servers)