

13

Κατανεμημένα Συστήματα

Ρολόγια
(Clocks)

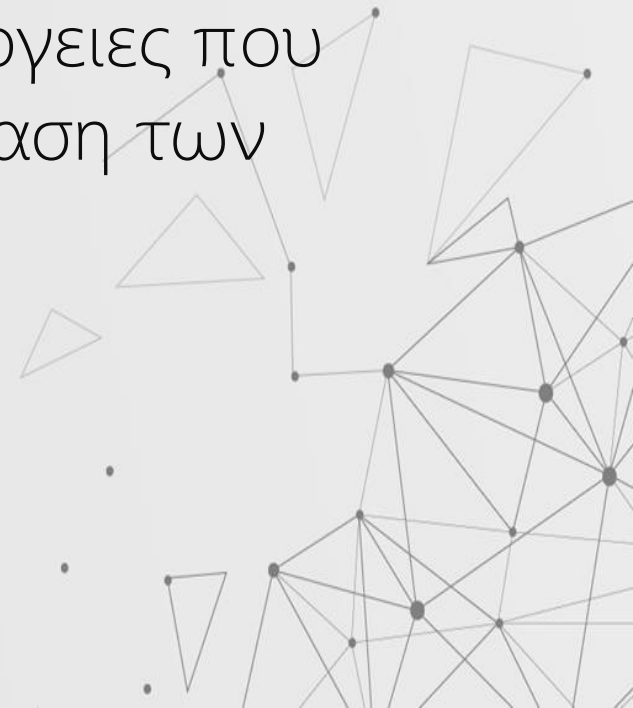


01

ΧΡΟΝΟΛΟΓΙΚΗ ΚΑΙ
(ΑΙΤΙΟ)ΛΟΓΙΚΗ ΣΕΙΡΑ

Τοπικά γεγονότα/συμβάντα

- Ορίζουμε ως γεγονός $e_{i.x}$ την x -οστή ενέργεια που έλαβε χώρα τοπικά στην διεργασία P_i .
- Το επίπεδο αφαίρεσης (το τι δηλαδή θεωρείται ως ένα «σημαντικό γεγονός») εξαρτάται από την εφαρμογή.
- Στην πράξη ενδιαφέρουν μόνο ενέργειες που αλλάζουν την (ουσιαστική) κατάσταση των διεργασιών που μας ενδιαφέρει να παρακολουθήσουμε.



Τοπική σειρά (;)

- Κάθε διεργασία P_i εκτελείται σειριακά.
- ; σημαίνει «έλαβε χώρα τοπικά αμέσως πριν το».
- ;...; σημαίνει «έλαβε χώρα τοπικά πριν το».

$$e_{i.x}; e_{i.y} \Rightarrow e_{i.x}; \dots; e_{i.y}$$

$$e_{i.x}; e' \quad e'; \dots; e_{i.y} \Rightarrow e_{i.x}; \dots; e_{i.y}$$

- Τα γεγονότα κάθε P_i είναι αυστηρά διατεταγμένα όταν:

$$\forall e_{i.x}, e_{j.x} : e_{i.x}; \dots; e_{j.x} \vee e_{j.x}; \dots; e_{i.x}$$

Χρονολογική σειρά ($\rightarrow$)

- $\rightarrow$ σημαίνει «έλαβε χώρα πριν το».
- Ορίζει την απόλυτη χρονική σειρά των γεγονότων στο κατανεμημένο σύστημα, με βάση έναν εξωτερικό παρατηρητή (ας υποθέσουμε ότι υπάρχει).
- Θεωρούμε ότι δεν υπάρχουν παράλληλα γεγονότα:

$$\forall e_{i.x}, e_{j.y} : e_{i.x} \rightarrow e_{j.y} \vee e_{j.y} \rightarrow e_{i.x}$$

- Συνέπεια τοπικής εκτέλεσης:

$$e_{i.x}; \dots; e_{i.y} \Leftrightarrow e_{i.x} \rightarrow e_{i.y}$$

- Μεταβατική ιδιότητα:

$$e_{i.x} \rightarrow e', e' \rightarrow e_{j.y} \Leftrightarrow e_{i.x} \rightarrow e_{j.y}$$

Αιτιολογική σειρά happened before $\mapsto$

- $\mapsto$ σημαίνει «προηγείται λογικά του».
- Καταγράφει την αιτιολογική σχέση των γεγονότων.
- Συνέπεια τοπικής εκτέλεσης:

$$e_{i.x}; \dots; e_{i.y} \Leftrightarrow e_{i.x} \mapsto e_{i.y}$$

- Συνέπεια επικοινωνίας:

$$e_{i.x} \equiv \text{snd}(m), e_{j.y} \equiv \text{rcv}(m) \Leftrightarrow e_{i.x} \mapsto e_{j.y}$$

- Μεταβατική ιδιότητα:

$$e_{i.x} \mapsto e', e' \mapsto e_{j.y} \Leftrightarrow e_{i.x} \mapsto e_{j.y}$$

- Η λογική σειρά σέβεται την χρονολογική σειρά

$$e_{i.x} \mapsto e_{j.y} \Rightarrow e_{i.x} \rightarrow e_{j.y}.$$



Παρατήρηση

- Αν $e_{i.x} \mapsto e_{j.y}$ γεγονότα της ίδιας διεργασίας, τότε το $e_{i.x}$ εκτελέστηκε τοπικά πριν το $e_{j.y}$.

$$e_{i.x} \mapsto e_{j.y}, i = j \Rightarrow e_{i.x}; \dots; e_{j.y}$$

- Αν $e_{i.x} \mapsto e_{j.y}$ γεγονότα από διαφορετικές διεργασίες, τότε συνδέονται μέσω της αποστολής/παραλαβής τουλάχιστον ενός μηνύματος.

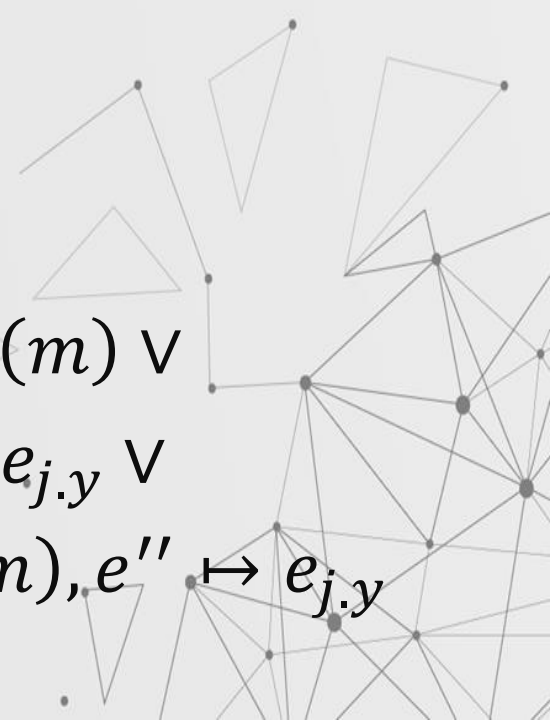
$$e_{i.x} \mapsto e_{j.y}, i \neq j \Rightarrow$$

$$e_{i.x} \equiv \text{snd}(m), e_{j.y} \equiv \text{rcv}(m) \vee$$

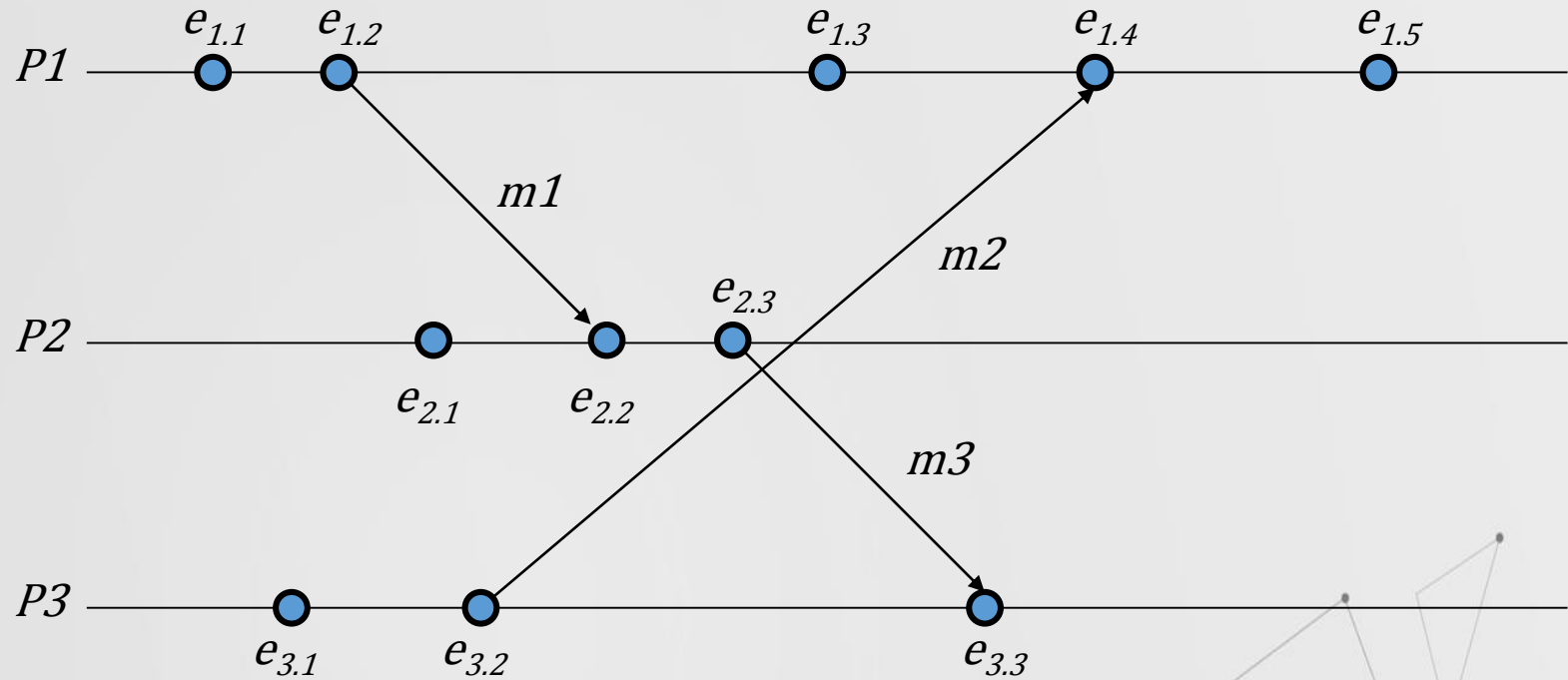
$$e_{i.x} \mapsto e', e' \equiv \text{snd}(m), e_{j.y} \equiv \text{rcv}(m) \vee$$

$$e_{i.x} \equiv \text{snd}(m), e' \equiv \text{rcv}(m), e' \mapsto e_{j.y} \vee$$

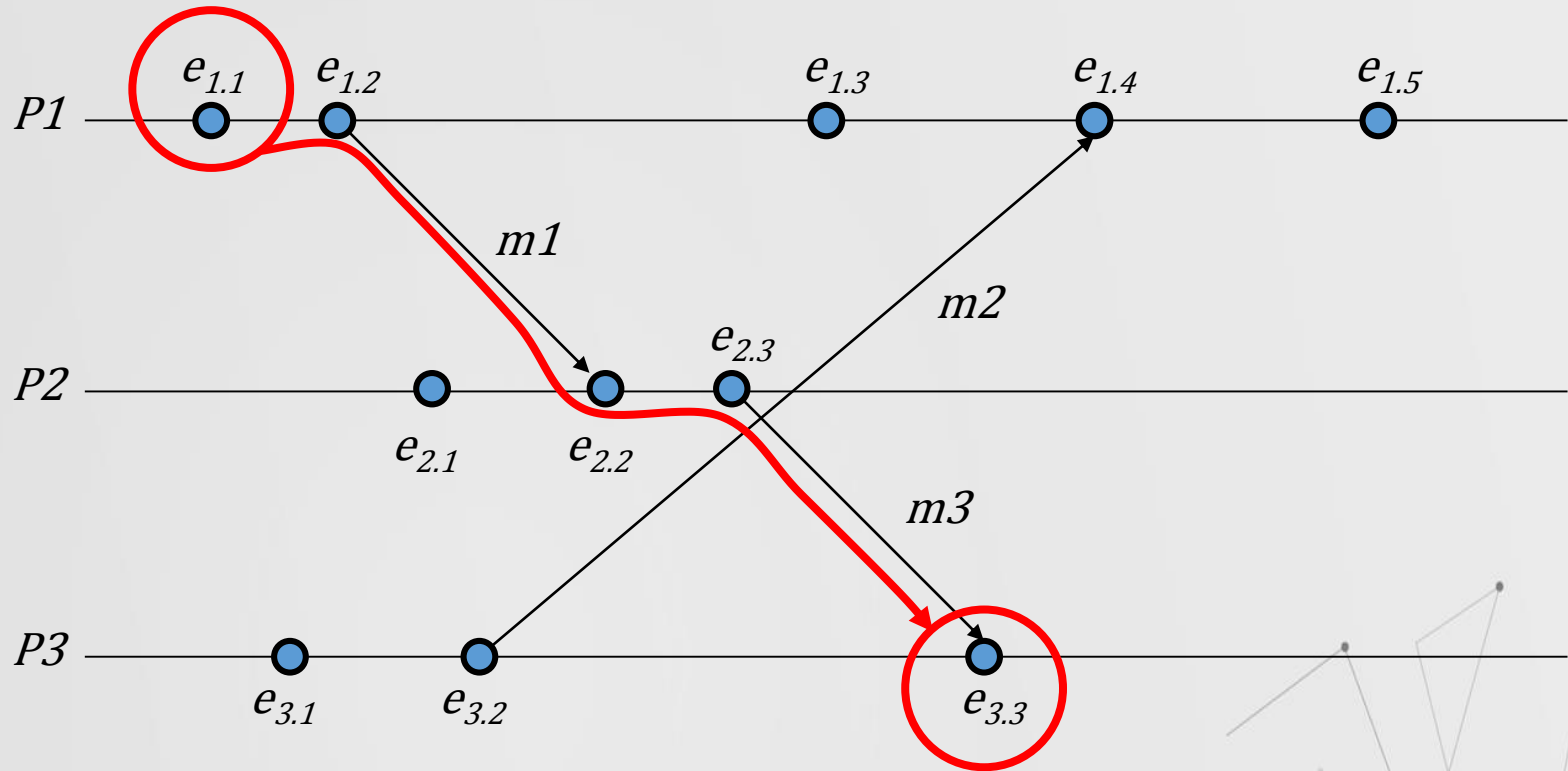
$$e_{i.x} \mapsto e', e' \equiv \text{snd}(m), e'' \equiv \text{rcv}(m), e'' \mapsto e_{j.y}$$



Αιτιολογική – Χρονολογική σειρά (1)



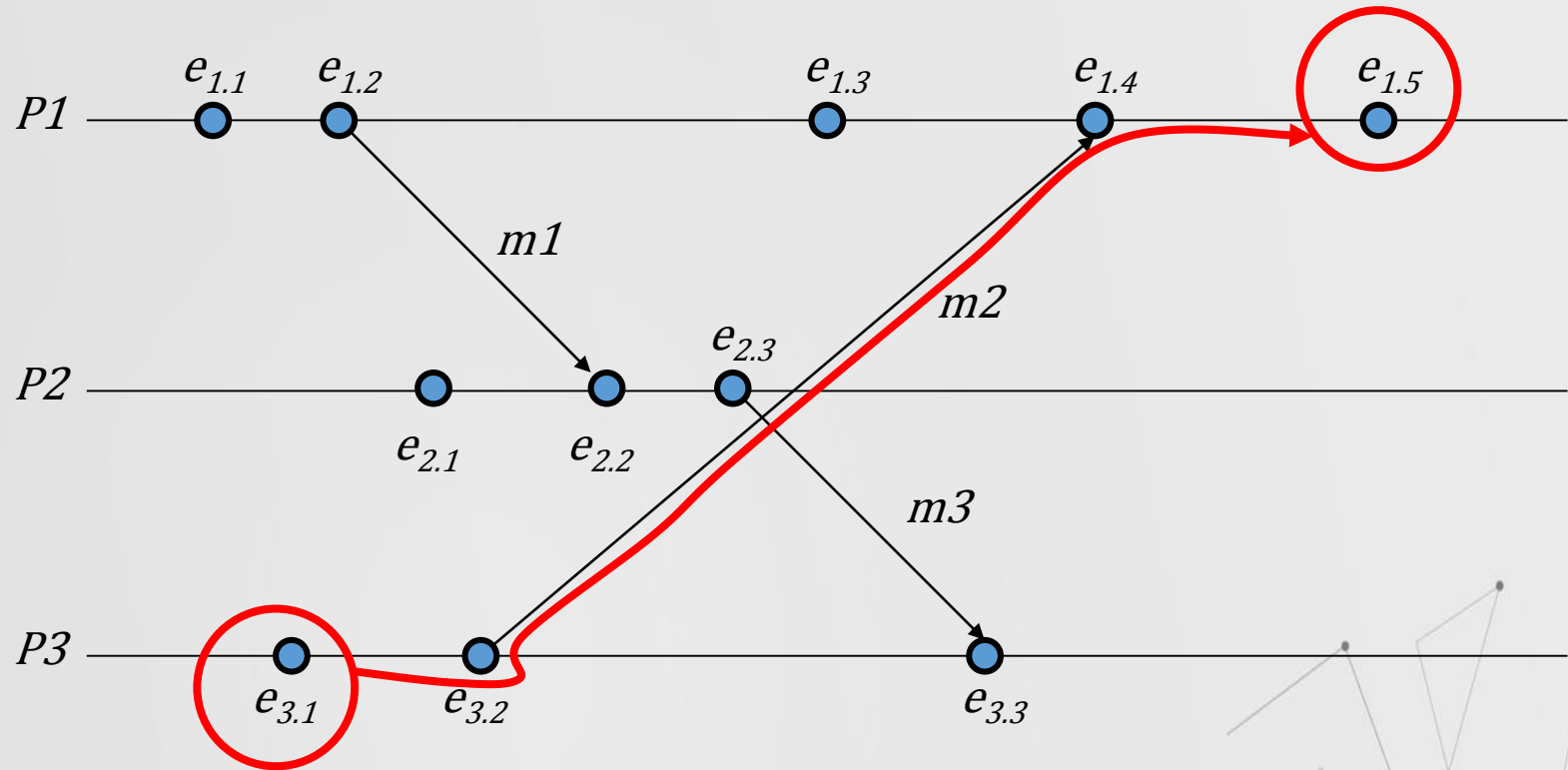
Αιτιολογική – Χρονολογική σειρά (2)



$e_{1.1} \mapsto e_{3.3} :$

$: e_{1.1} \mapsto e_{1.2}, e_{1.2} \mapsto e_{2.2}, e_{2.2} \mapsto e_{2.3}, e_{2.3} \mapsto e_{3.3}$

Αιτιολογική – Χρονολογική σειρά (3)



$e_{3.1} \mapsto e_{1.5} :$

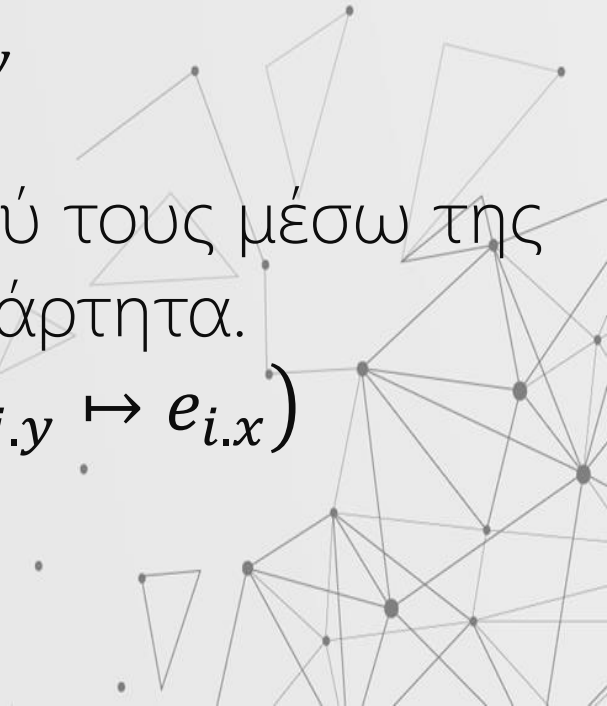
$: e_{3.1} \mapsto e_{3.2}, e_{3.2} \mapsto e_{1.4}, e_{1.4} \mapsto e_{1.5}$

Λογική ανεξαρτησία

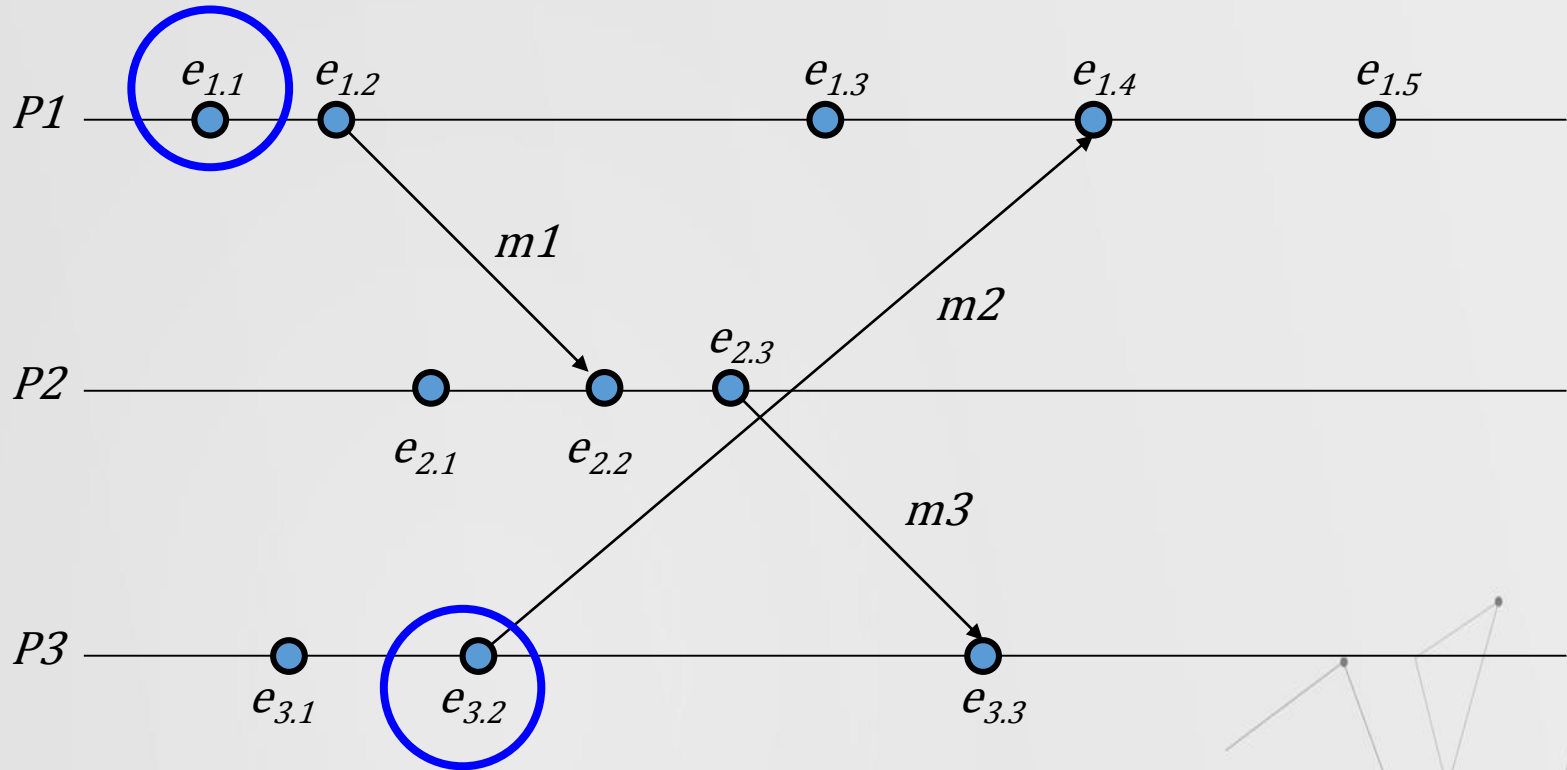
- Η σχέση $\mapsto$ ορίζει μια μερική διάταξη στο σύνολο των γεγονότων του κατανεμημένου συστήματος.
- Η σχέση $\mapsto$ είναι υποσύνολο της σχέσης $\rightarrow$.
- Το ότι ένα γεγονός έλαβε χώρα πριν από ένα άλλο δεν συνεπάγεται ότι προηγείται λογικά από αυτό:

$$e_{i.x} \rightarrow e_{j.y} \not\Rightarrow e_{i.x} \mapsto e_{j.y}$$

- Γεγονότα που δεν συνδέονται μεταξύ τους μέσω της σχέσης $\mapsto$, ονομάζονται λογικά ανεξάρτητα.

$$e_{i.x} || e_{j.y} \equiv \neg(e_{i.x} \mapsto e_{j.y}) \wedge \neg(e_{j.y} \mapsto e_{i.x})$$


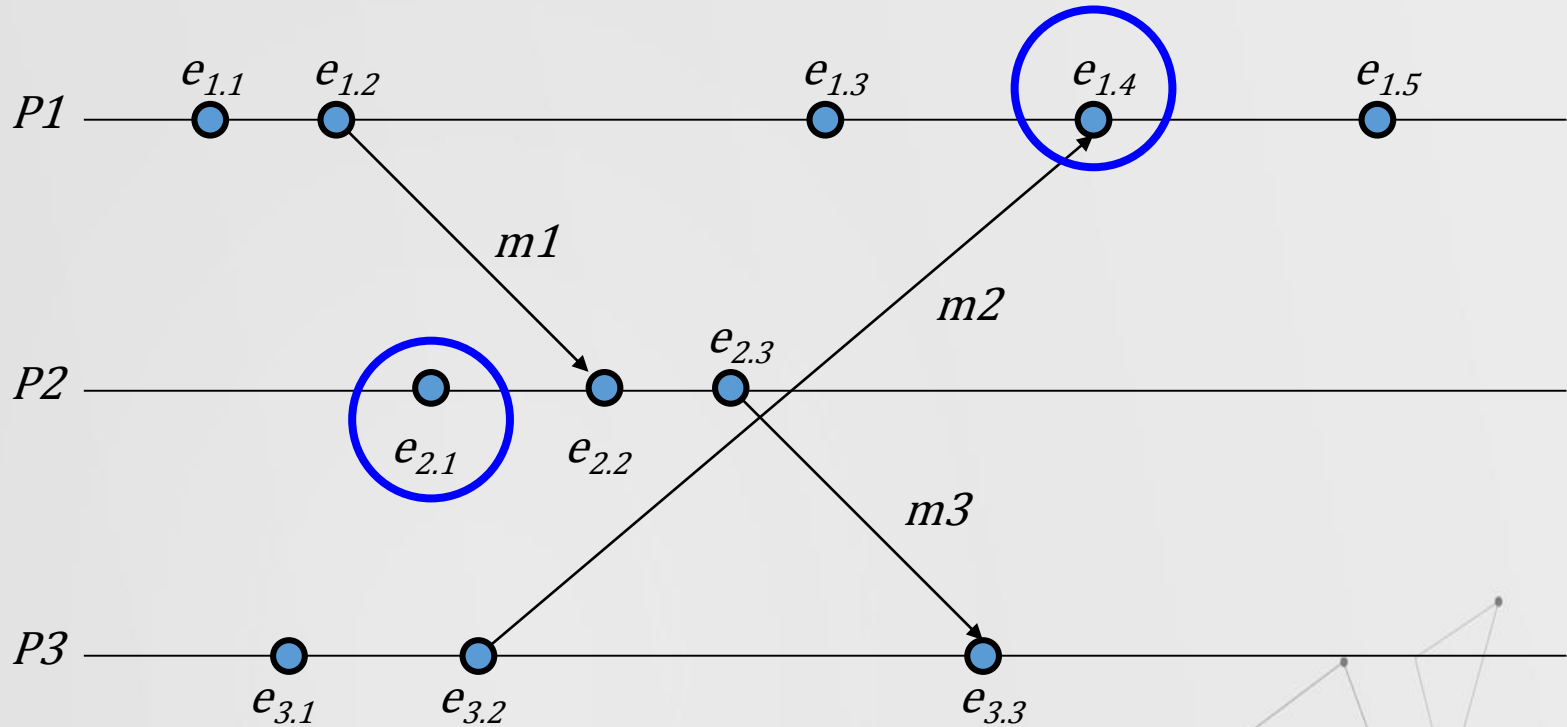
Λογική ανεξαρτησία



$e_{1.1} \parallel e_{3.2}$ παρ' ότι $e_{1.1} \rightarrow e_{3.2}$



Λογική ανεξαρτησία



$e_{2.1} \parallel e_{1.4}$ παρ' ότι $e_{2.1} \rightarrow e_{1.4}$

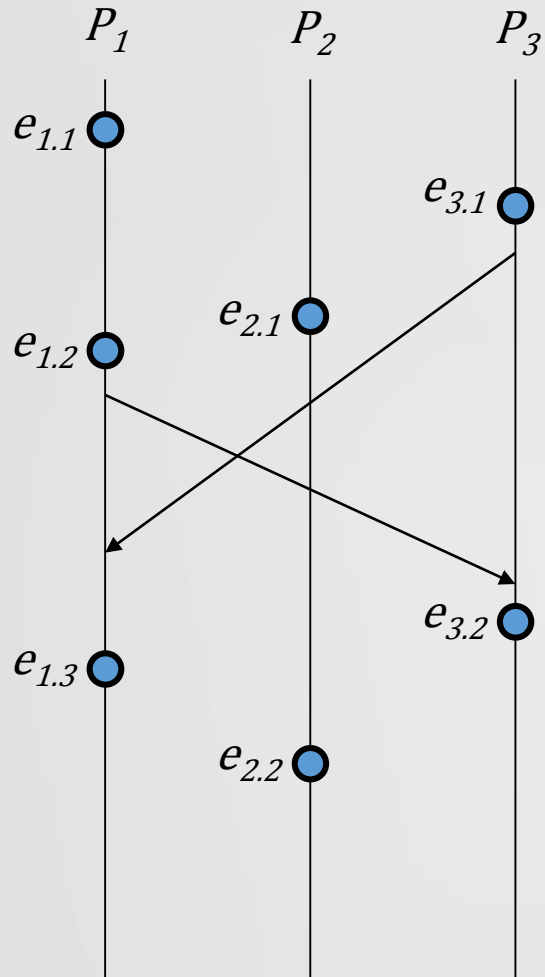


Χρονική και αιτιολογική σειρά

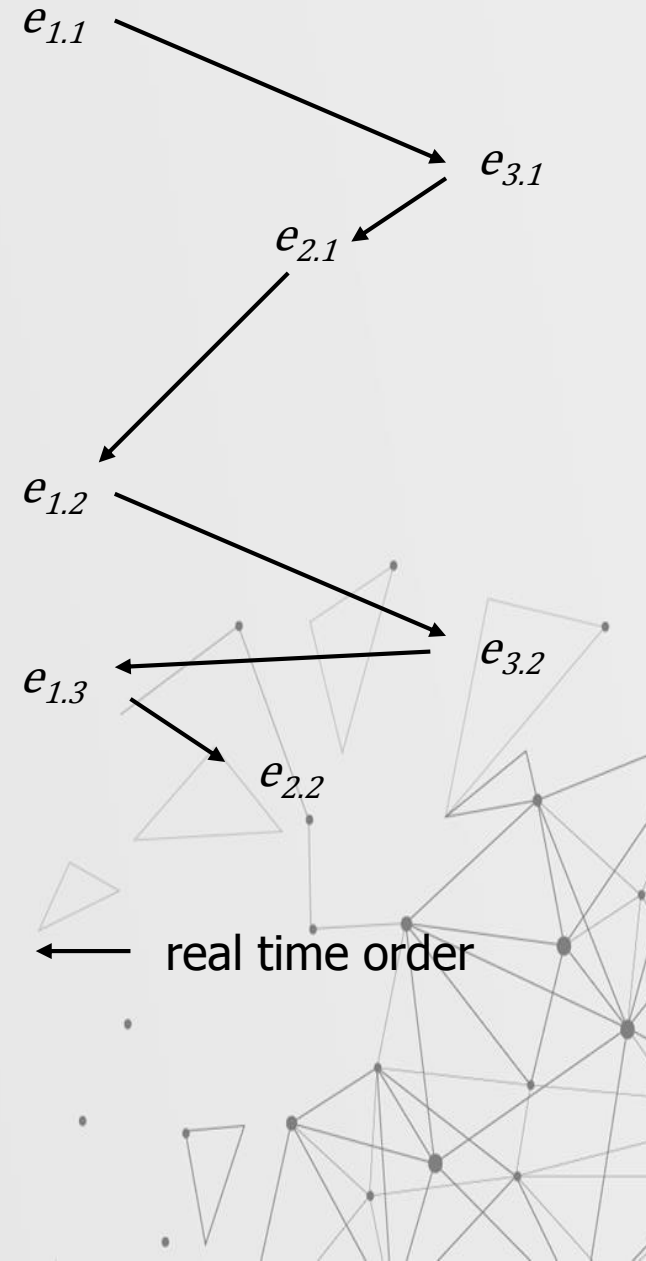
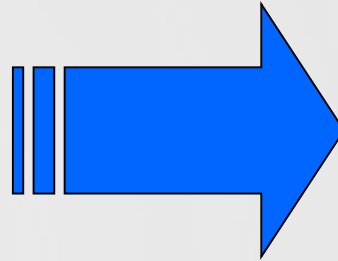
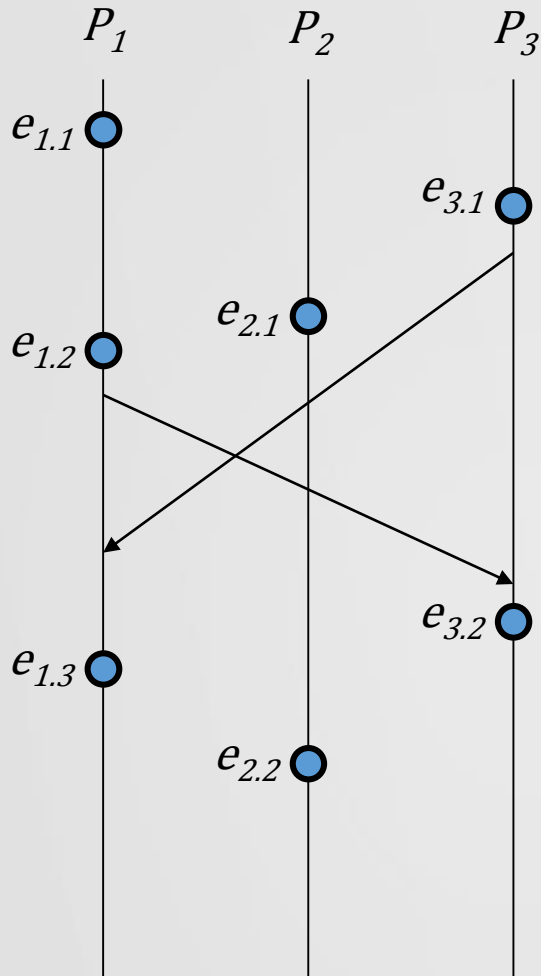
- Χρονική σειρά: δίνει απάντηση στο ερώτημα κατά πόσο ένα γεγονός έλαβε χώρα πριν από ένα άλλο.
- Αιτιολογική σειρά: δίνει απάντηση στο κατά πόσο ένα γεγονός (πιθανόν να) προκάλεσε ένα άλλο.
- Η αιτιολογική σειρά σέβεται την χρονική σειρά.
- Η αιτιολογική σειρά δεν ταυτίζεται με την χρονική σειρά.



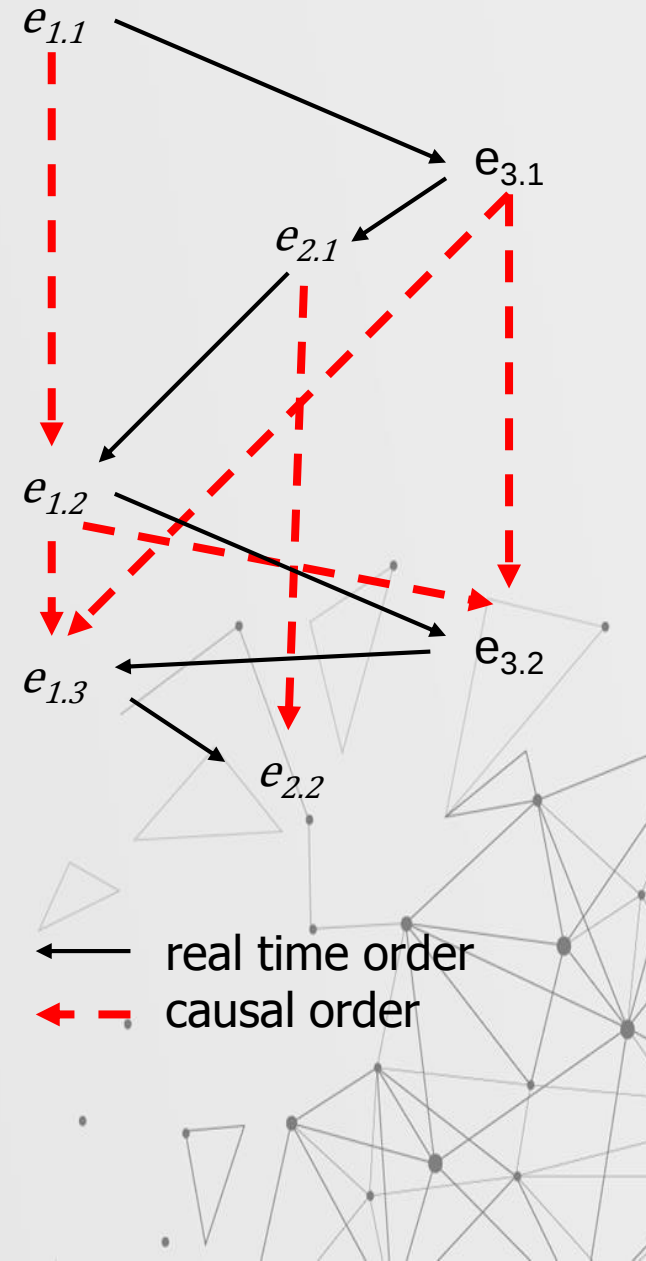
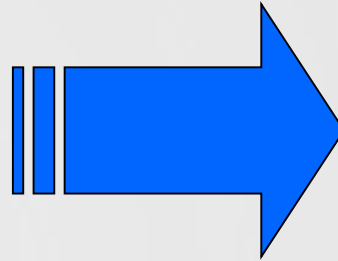
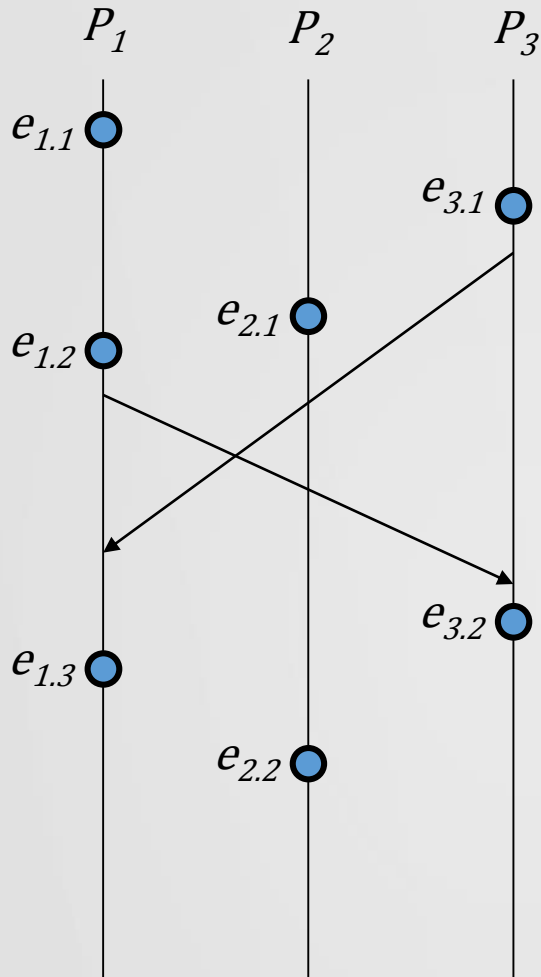
Χρονική και αιτιολογική σειρά



Χρονική και αιτιολογική σειρά

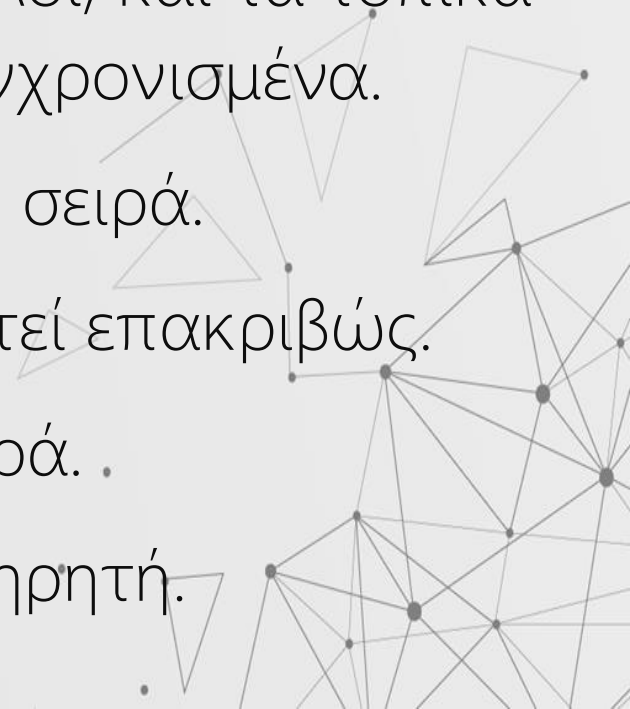


Χρονική και αιτιολογική σειρά



Κατανεμημένη εκτέλεση

- Κάποιες φορές θέλουμε να γνωρίζουμε την πραγματική χρονική σειρά με την οποία έλαβαν χώρα τα γεγονότα.
- Όμως, στην οριακή περίπτωση, αυτό είναι αδύνατο.
- Δεν υπάρχει αντικειμενικός εξωτερικός παρατηρητής
- Οι διεργασίες δεν διαθέτουν κοινό ρολόι, και τα τοπικά ρολόγια των διεργασιών δεν είναι συγχρονισμένα.
- Συχνά, αρκεί να γνωρίζουμε τη λογική σειρά.
- Η λογική σειρά μπορεί να προσδιοριστεί επακριβώς.
- Χωρίς να γνωρίζουμε την χρονική σειρά. .
- Χωρίς την ύπαρξη εξωτερικού παρατηρητή.



Δύο διαφορετικές προσεγγίσεις

- **Φυσικά ρολόγια:** συγχρονισμός των φυσικών ρολογιών των διεργασιών με ικανοποιητική ακρίβεια, και (ιδανικά) σύμφωνα με τον «πραγματικό» χρόνο.
- **Λογικά ρολόγια:** διαχείριση τοπικών μετρητών για τα γεγονότα που λαμβάνουν χώρα σε κάθε διεργασία.
 - χωρίς να γίνεται καταγραφή του πραγματικού χρόνου.
- Και στις δύο περιπτώσεις, αναθέτουμε στα γεγονότα χρονοσφραγίδες (timestamps) με βάση τις οποίες τα κατατάσσουμε σε σχέση με τα υπόλοιπα γεγονότα.
 - η κατάταξη μπορεί π.χ. να γίνει από μια διεργασία που λαμβάνει μηνύματα ειδοποίησης για κάθε γεγονός.

Χρονοσφραγίδες

- Γνωστές και ως timestamps.
 - The UNIX time stamp is a way to track time as a running total of seconds. This count starts at the Unix Epoch on January 1st, 1970 at UTC.
 - Therefore, the UNIX time stamp is merely the number of seconds between a particular date and the Unix Epoch.
 - This point in time technically does not change no matter where you are located on the globe.
 - This is very useful to computer systems for tracking and sorting dated information in dynamic and distributed applications both online and client side.
- 



02

ΦΥΣΙΚΑ ΡΟΛΟΓΙΑ

Χρόνος και ρολόγια

- Κάθε διεργασία P_i διαθέτει τοπικό ρολόι $C_i()$.
- $C_i(t)$ η τιμή του ρολογιού την χρονική στιγμή t .
- Το ρολόι δεν είναι εντελώς σωστό: $C_i(t) \neq t$.
- Πρόβλημα ακρίβειας (accuracy): το ρολόι μετράει τον χρόνο με διακριτό τρόπο.
- Πρόβλημα απόκλισης (drift): το ρολόι μπορεί να «χάνει/κερδίζει» στον χρόνο.



Επιθυμητή συμπεριφορά ρολογιού

- Ένα ρολόι δεν επιτρέπεται να πάει «προς τα πίσω».

$$t > t' \Rightarrow C_i(t) \geq C_i(t')$$

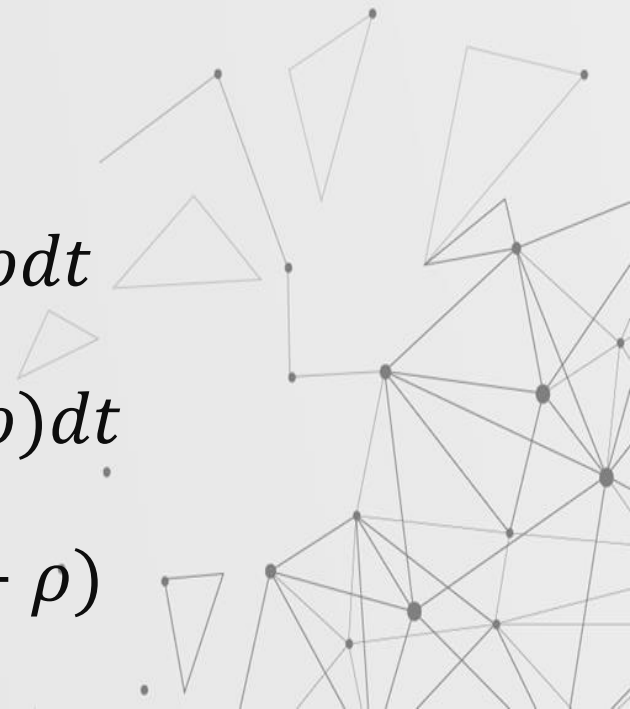
- Ένα ρολόι δεν πραγματοποιεί «απρόβλεπτα» άλματα, δηλαδή έχει εγγυημένο άνω όριο ρυθμού απόκλισης ρ .

- Για $dt = t' - t$, $dC = C(t') - C(t)$:

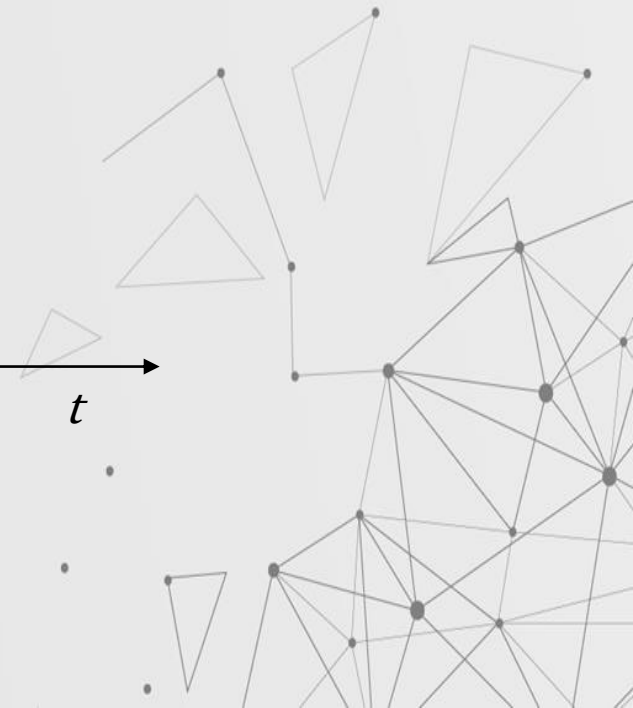
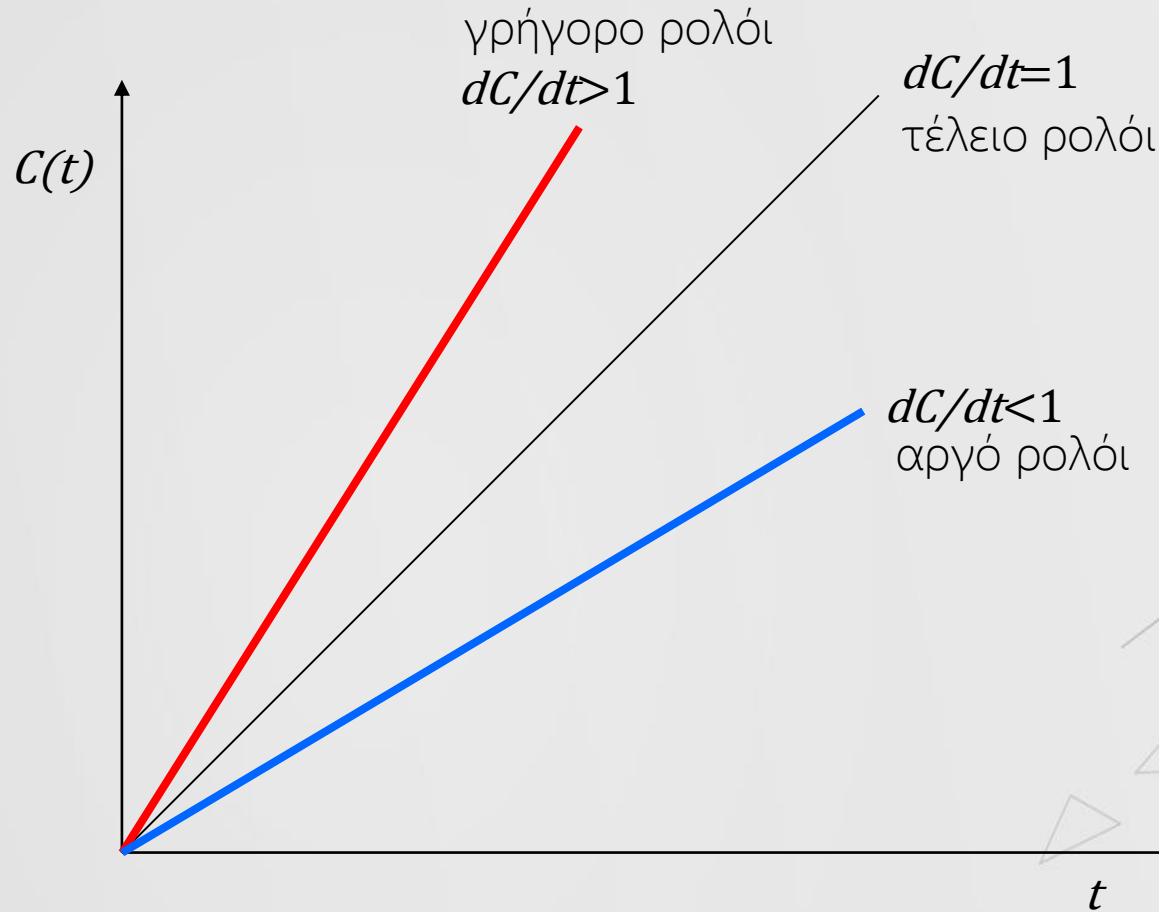
$$dt - \rho dt \leq dC \leq dt + \rho dt$$

$$(1 - \rho)dt \leq dC \leq (1 + \rho)dt$$

$$(1 - \rho) \leq dC/dt \leq (1 + \rho)$$

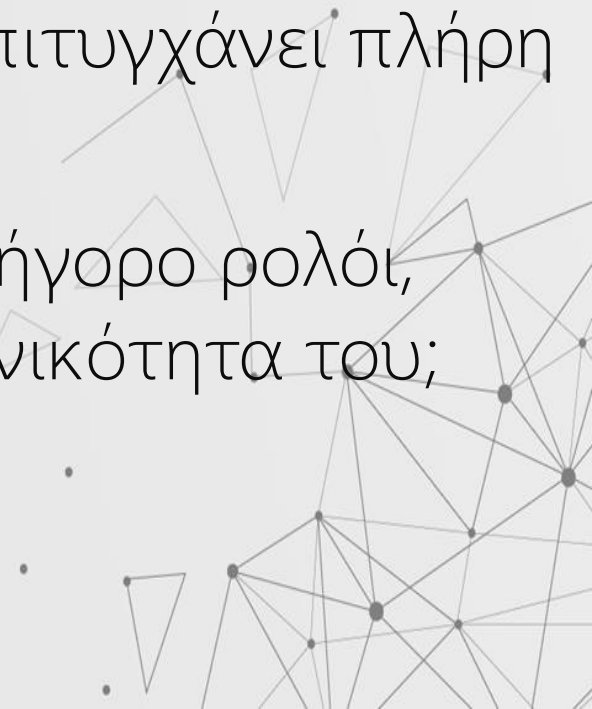


Επιθυμητή συμπεριφορά ρολογιού

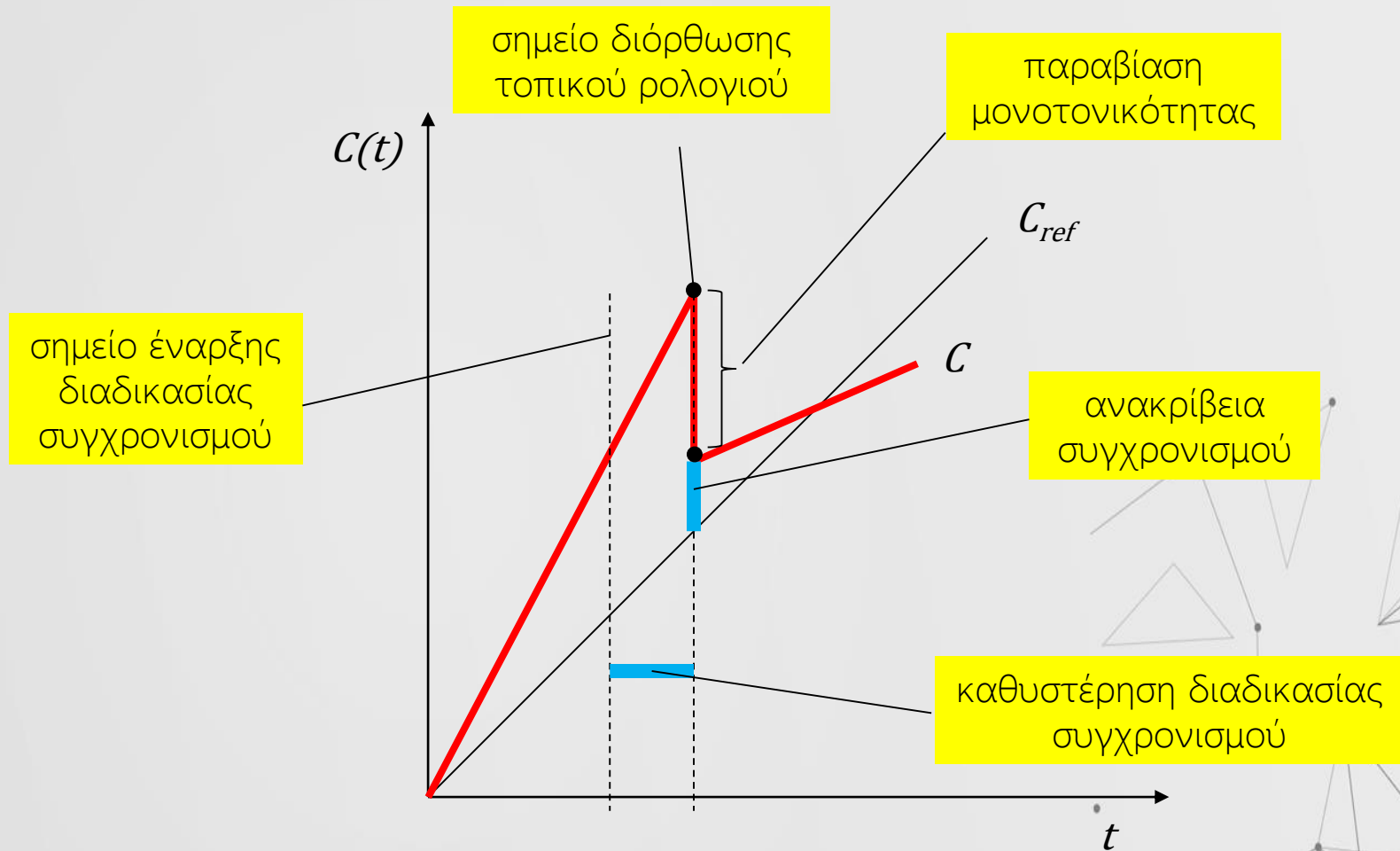


Συγχρονισμός φυσικών ρολογιών

- Ιδανικά: $\forall P_i, P_j : C_i(t) = C_j(t) = t$.
- Οι διεργασίες πρέπει να συγχρονίσουν τα ρολόγια τους – και αυτό πρέπει να γίνεται περιοδικά.
- Σημείωση: Η διαδικασία συγχρονισμού απαιτεί κάποιο (μικρό;) χρονικό διάστημα για να ολοκληρωθεί, και πιθανώς να μην επιτυγχάνει πλήρη συγχρονισμό.
- Σημείωση: πως διορθώνουμε ένα γρήγορο ρολόι, χωρίς να παραβιάσουμε την μονοτονικότητά του;



Συγχρονισμός φυσικών ρολογιών



Είδη συγχρονισμού

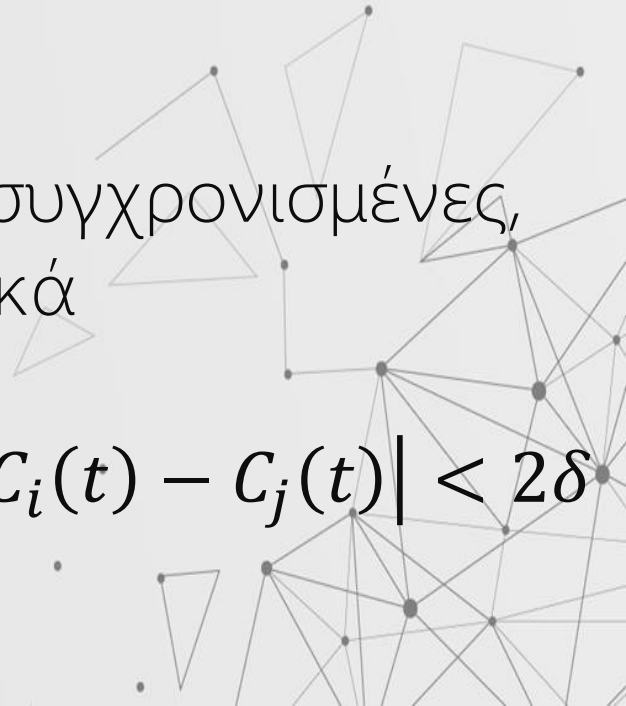
- Εξωτερικός συγχρονισμός: με ρολόι αναφοράς:

$$\forall P_i : |C_{ref}(t) - C_i(t)| < \delta$$

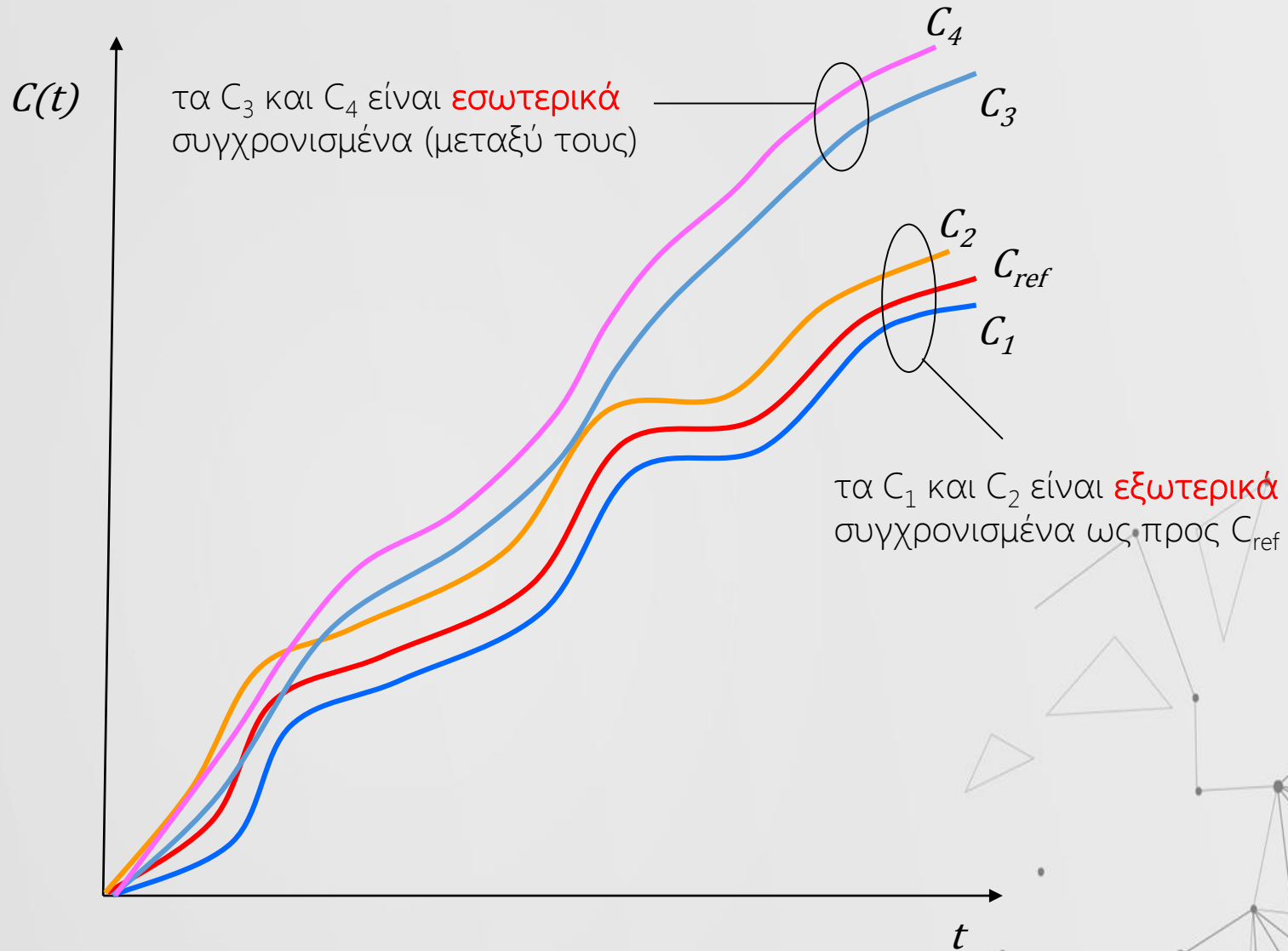
- Εσωτερικός συγχρονισμός: χωρίς ρολόι αναφοράς:

$$\forall P_i, P_j : |C_i(t) - C_j(t)| < \delta$$

- Αν δύο διεργασίες είναι εξωτερικά συγχρονισμένες, συνεπάγεται πως είναι και εσωτερικά συγχρονισμένες.

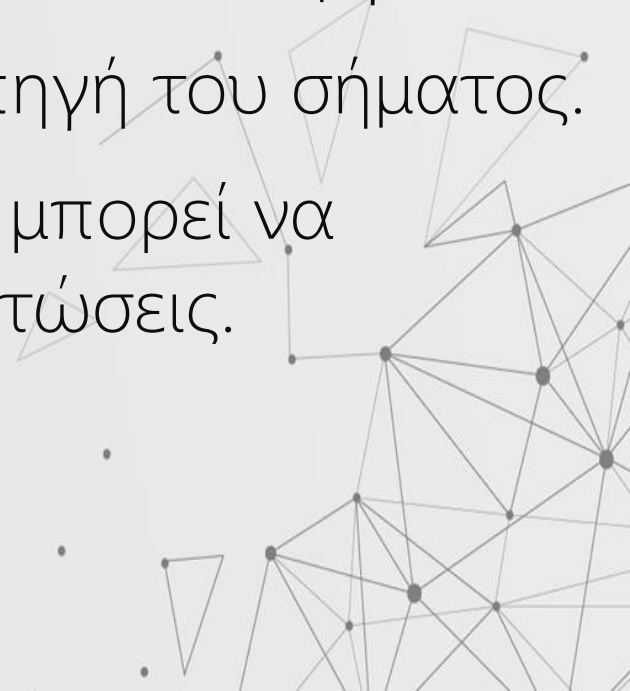
$$\forall P_i : |C_{ref}(t) - C_i(t)| < \delta \Rightarrow \forall P_i, P_j : |C_i(t) - C_j(t)| < 2\delta$$


Είδη συγχρονισμού



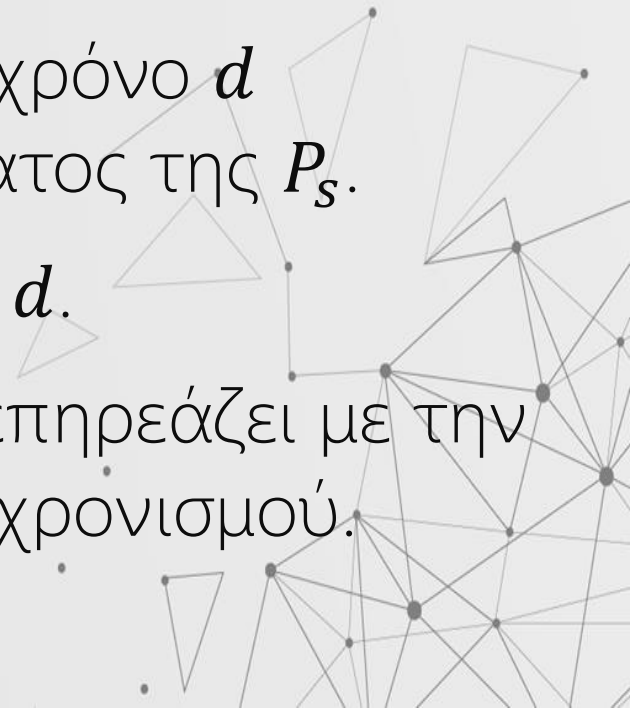
Ασύρματοι δέκτες ώρας ακριβείας

- Δέκτες UTC Coordinated Universal Time (UTC).
- Δέκτες GPS.
- Ένας Η/Υ μπορεί να χρησιμοποιεί τέτοιους δέκτες για να συγχρονίζει περιοδικά το ρολόι του.
- Δεν απαιτείται επικοινωνία ανάμεσα στις διεργασίες.
- Υπάρχει όμως επικοινωνία με την πηγή του σήματος.
- Τέτοιοι δέκτες είναι ακριβοί και δεν μπορεί να χρησιμοποιηθούν σε όλες τις περιπτώσεις.



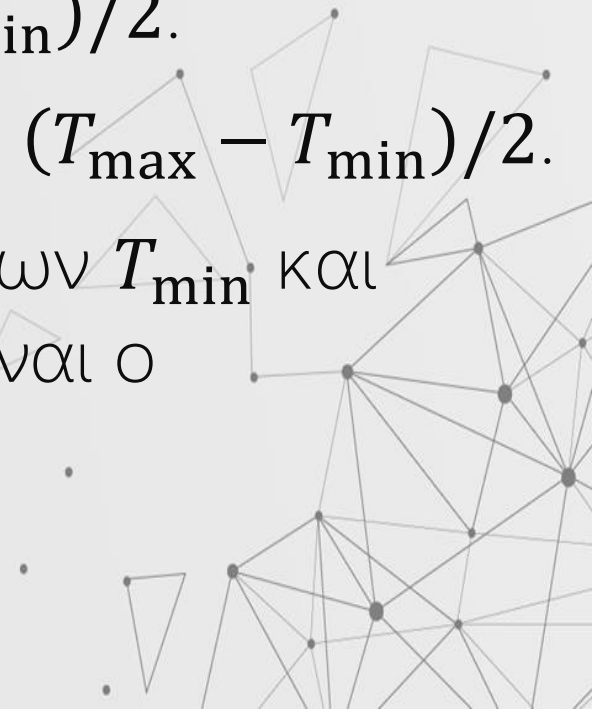
Συγχρονισμός με μονομερή ανακοίνωση

- Διακεκριμένη διεργασία P_s ως σημείο αναφοράς.
- Η P_s ανακοινώνει την τιμή του ρολογιού της με ένα μήνυμα προς τις ενδιαφερόμενες διεργασίες.
- Κάθε άλλη διεργασία P_i συγχρονίζει το ρολόι της με βάση την τιμή που έλαβε από την P_s .
- Πρόβλημα: η P_i δεν γνωρίζει πόσο χρόνο d χρειάστηκε η μετάδοση του μηνύματος της P_s .
- Πρέπει να γίνει μια εκτίμηση για το d .
- Η ακρίβεια της εκτίμησης για το d επηρεάζει με την σειρά της και την ακρίβεια του συγχρονισμού.

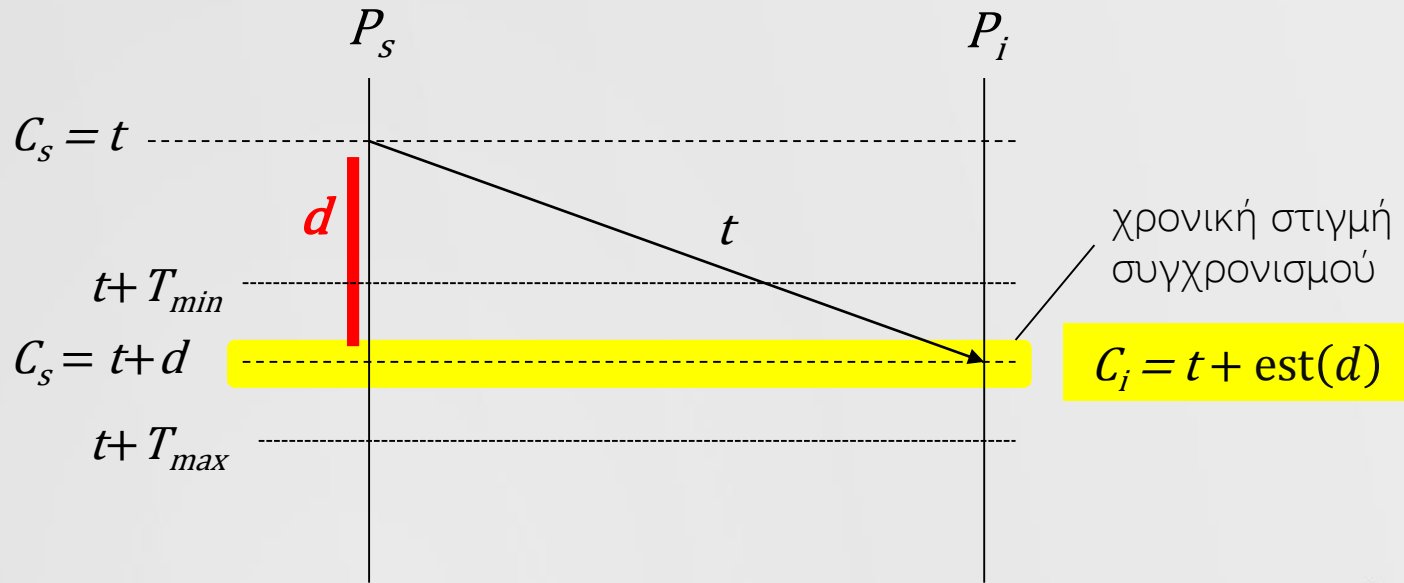


Εκτίμηση της καθυστέρησης μετάδοσης

- $T_{\max}$: άνω όριο καθυστέρησης μετάδοσης.
- $T_{\min}$: κάτω όριο καθυστέρησης μετάδοσης.
- Εκτίμηση της P_i : $\text{est}(d) = T_{\max}$ ή $\text{est}(d) = T_{\min}$.
- Ανακρίβεια συγχρονισμού: $|C_s - C_i| \leq T_{\max} - T_{\min}$.
- Εκτίμηση της P_i : $\text{est}(d) = (T_{\max} + T_{\min})/2$.
- Ανακρίβεια συγχρονισμού: $|C_s - C_i| \leq (T_{\max} - T_{\min})/2$.
- Όσο πιο κοντά βρίσκονται οι τιμές των $T_{\min}$ και $T_{\max}$, τόσο πιο εγγυημένα ακριβής είναι ο συγχρονισμός.



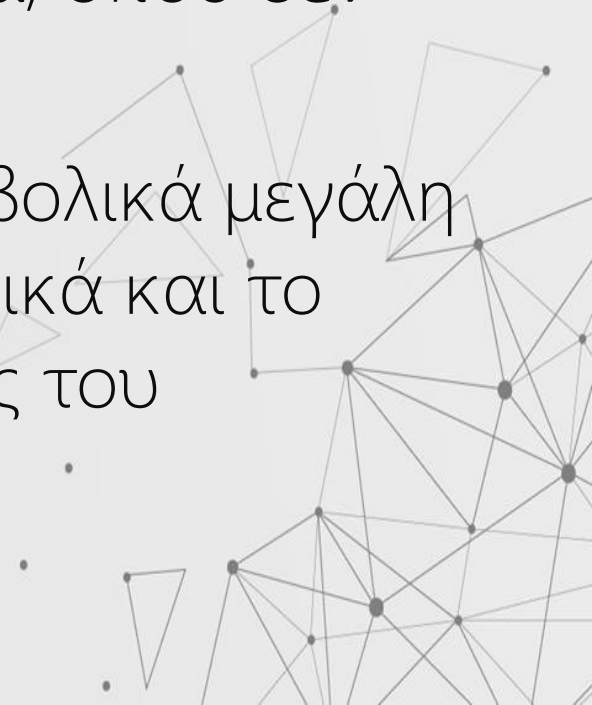
Εκτίμηση της καθυστέρησης μετάδοσης



$\text{est}(d)$	Worst d	$ C_s - C_i $
T_{min}	T_{max}	$T_{max} - T_{min}$
T_{max}	T_{min}	$T_{max} - T_{min}$
$(T_{max} + T_{min})/2$	T_{max}, T_{min}	$(T_{max} - T_{min})/2$

Στην πράξη

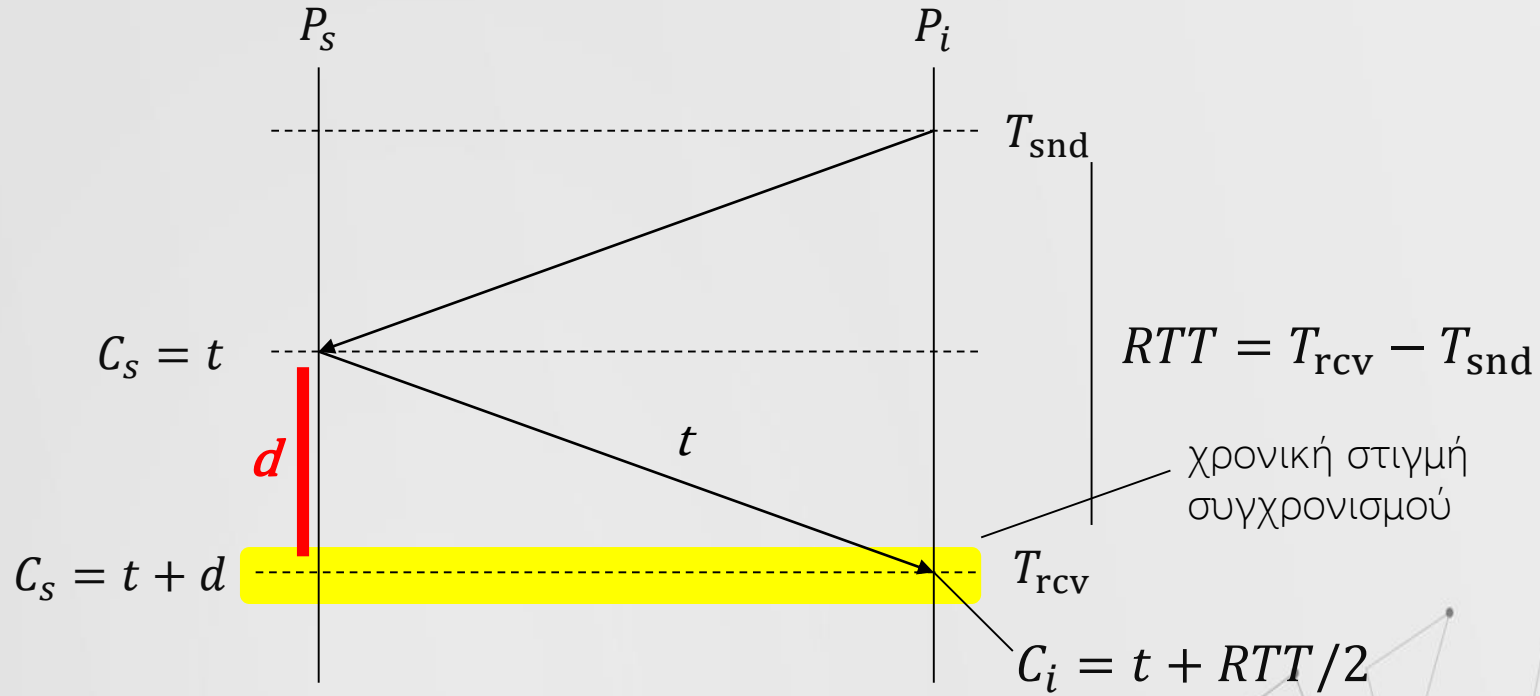
- Σε κάποια συστήματα, τα $T_{\min}$ και $T_{\max}$ μπορεί να προσεγγιστούν συντηρητικά.
 - με βάση τα χαρακτηριστικά του φυσικού δικτύου,
 - το χρόνο επεξεργασίας σε έναν τυπικό κόμβο.
- Τι γίνεται σε ένα ασύγχρονο σύστημα, όπου δεν μπορεί να προκαθοριστεί το $T_{\max}$;
- Μπορεί να χρησιμοποιηθεί μια υπερβολικά μεγάλη τιμή, αυτό όμως μεγαλώνει υπερβολικά και το εγγυημένο άνω όριο της ανακρίβειας του συγχρονισμού.
 - δεν είναι ιδιαίτερα χρήσιμο.



Συγχρονισμός με αίτηση (αλγόριθμος Cristian)

- Η P_i στέλνει αίτηση στην P_s .
- Η P_s απαντά με την τιμή του ρολογιού της.
- Η P_i σημειώνει την χρονική στιγμή αποστολής της αίτησης και την χρονική στιγμή παραλαβής της απάντησης, και υπολογίζει $RTT = T_{rcv} - T_{snd}$.
 - η μέτρηση γίνεται αποκλειστικά με βάση το ρολόι της P_i .
 - ακριβής αν το ρολόι δεν έχει μεγάλο ρυθμό απόκλισης.
- Η εκτίμηση για το d είναι $est(d) = RTT/2$.
- Η P_i διορθώνει το ρολόι της $C_i = t + RTT/2$.
- Η μέγιστη διαφορά συγχρονισμού είναι $\frac{RTT}{2} - T_{min}$.
- Δεν χρειάζεται να γνωρίζουμε το T_{max} .

Συγχρονισμός με αίτηση (αλγόριθμος Cristian)

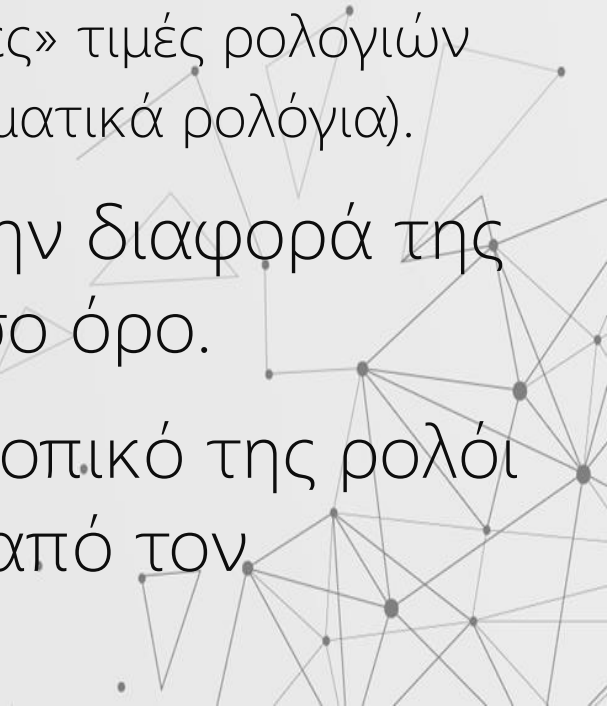


extremes for d	$ C_s - C_i = d - RTT/2 $
$d = T_{\min}$	$ T_{\min} - RTT/2 = RTT/2 - T_{\min} $
$d = RTT - T_{\min}$	$ RTT - T_{\min} - RTT/2 = RTT/2 - T_{\min} $

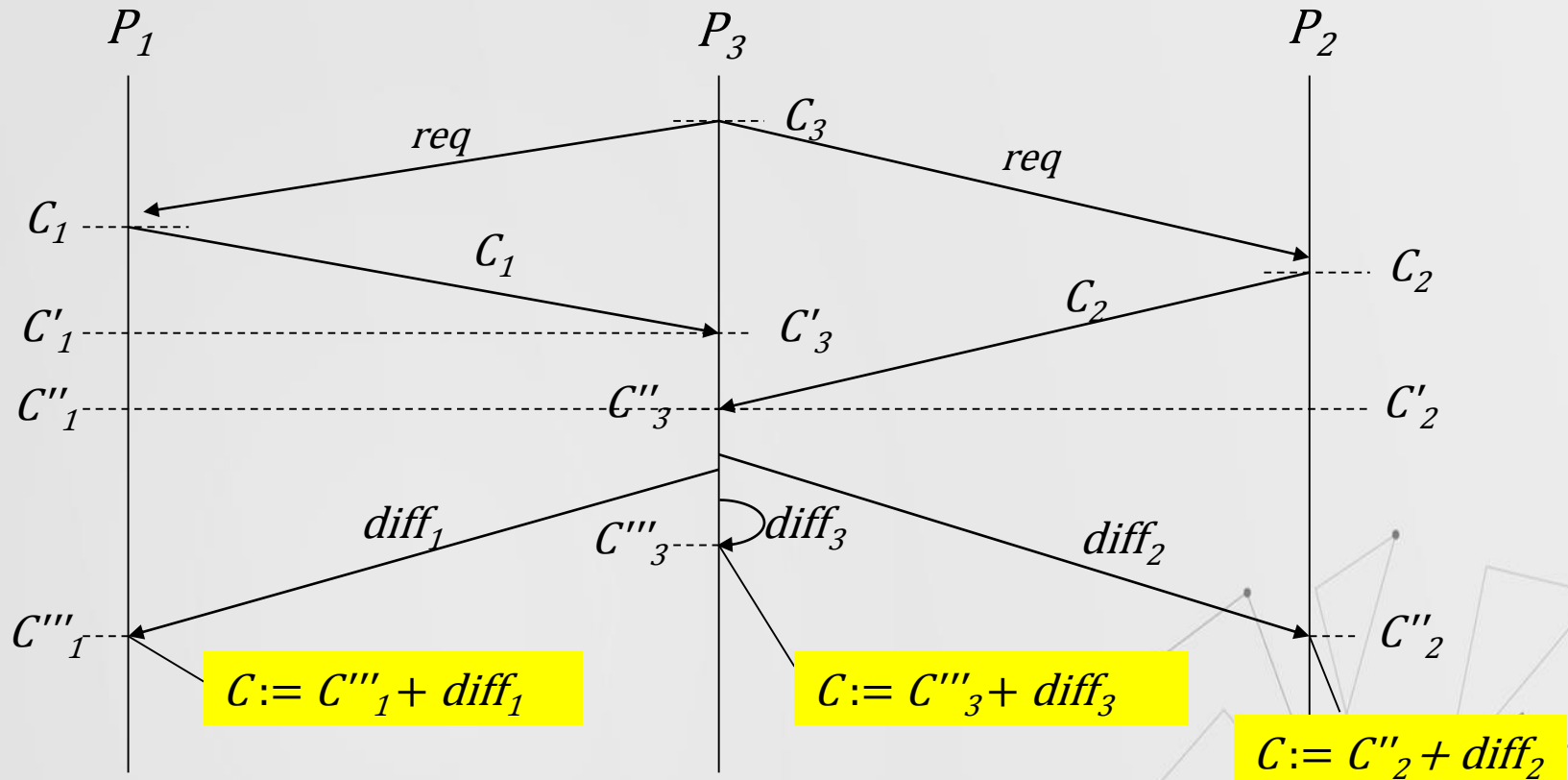
Παρατήρηση

- Υποθέτουμε ότι ο χρόνος που χρειάζεται η αίτηση για να φτάσει από την P_i στην P_s είναι (περίπου) ίσος με τον χρόνο που χρειάζεται η απάντηση για να φτάσει από την P_s στην P_i .
- Αυτό είναι μια αρκετά λογική υπόθεση που ισχύει με μεγάλη πιθανότητα στις περισσότερες περιπτώσεις.
 - όμως μπορεί να μην ισχύει.
- Το RTT μπορεί να λαμβάνει διαφορετική τιμή για κάθε μέτρηση (προσπάθεια συγχρονισμού).
- Η P_i πρέπει να προσαρμόζει δυναμικά την περίοδο συγχρονισμού της με την διεργασία P_s .
 - με βάση την επιθυμητή μέγιστη απόκλιση.

Εσωτερικός συγχρονισμός σε ομάδα (αλγόριθμος Gusella & Zatti – Berkeley)

- Μια από τις διεργασίες λειτουργεί ως συντονιστής.
 - Χρησιμοποιεί την μέθοδο Cristian για να υπολογίσει την τρέχουσα τιμή του ρολογιού κάθε διεργασίας P_i .
 - Σχηματίζει τον μέσο όρο.
 - πιθανώς αγνοώντας απαντήσεις που καθυστερούν πάνω από ένα χρονικό όριο καθώς και τυχόν «ακραίες» τιμές ρολογιών (για να υπάρχει ανοχή σε βλάβες ή ελαττωματικά ρολόγια).
 - Ενημερώνει κάθε διεργασία P_i για την διαφορά της τιμής του ρολογιού της από τον μέσο όρο.
 - Κάθε διεργασία P_i προσαρμόζει το τοπικό της ρολόι ανάλογα με την τιμή που λαμβάνει από τον συντονιστή.
- 

Αλγόριθμος Gusella & Zatti – Berkeley



$$\text{est}(C'_1'') = C_1 + (C'_3 - C_3)/2 + (C''_3 - C'_3)$$

$$\text{est}(C'_2) = C_2 + (C''_3 - C_3)/2$$

$$\text{avg} = (\text{est}(C'_1'') + \text{est}(C'_2) + C''_3)/3$$

$$diff_1 = \text{avg} - \text{est}(C'_1'')$$

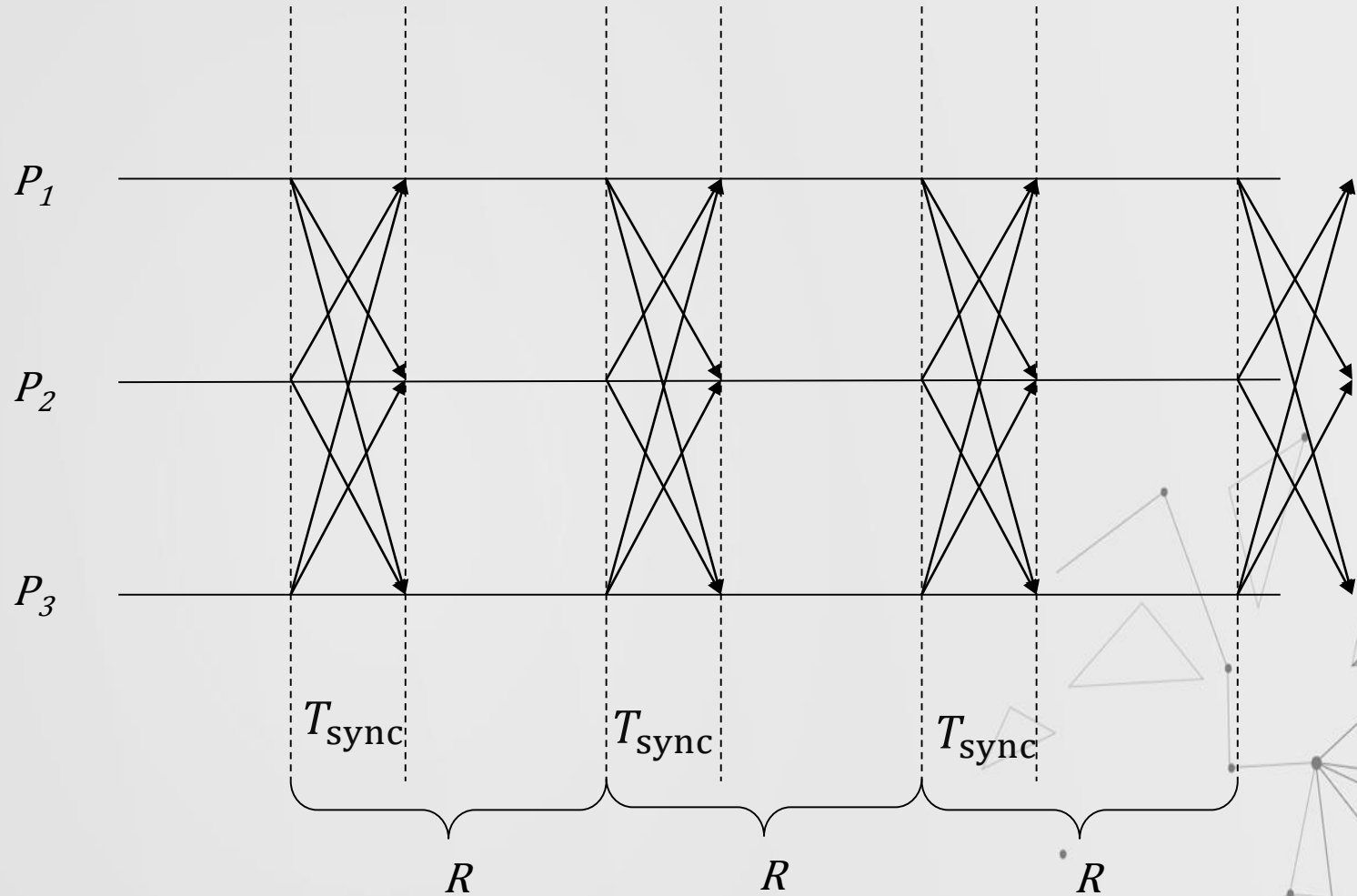
$$diff_2 = \text{avg} - \text{est}(C'_2)$$

$$diff_3 = \text{avg} - C''_3$$

Συμμετρικοί αλγόριθμοι

- Σύγχρονο σύστημα, γνωστό $T_{\max}$.
- Κάθε P_i στέλνει σε όλες τις άλλες διεργασίες την τιμή του ρολογιού της, π.χ., με ένα multicast/broadcast.
- Συλλέγει τις τιμές των υπολοίπων (περιμένει T_{sync}).
- Το T_{sync} είναι συνάρτηση του $T_{\max}$ και του δ .
- Υπολογίζει την «σωστή» ώρα (με βάση μια κοινή συνάρτηση), και προσαρμόζει το ρολόι της.
- Αυτό γίνεται περιοδικά, κάθε R μονάδες χρόνου.
- Το R είναι συνάρτηση του ρυθμού απόκλισης ρ , της μέγιστης απόκλισης ρολογιών δ , και του T_{sync} .

Συμμετρικοί αλγόριθμοι



Αλγόριθμος Lamport για $N \geq 3f + 1$ διεργασίες με f βυζαντινές βλάβες

- Επιτρέπονται το πολύ f βυζαντινές βλάβες.
- Κάποιες διεργασίες μπορεί να στέλνουν λάθος τιμές.
- Όταν η P_i λάβει t_k από την P_k , κάνει μια εκτίμηση $\text{est}_i(t_k)$ της τρέχουσας τιμής του ρολογιού της P_k .
- Βλ. μέθοδο συγχρονισμού με μονομερή ανακοίνωση.
- Αν $|C_i - \text{est}_i(t_k)| > \delta$, θεωρείται ότι η P_k παρουσίασε βλάβη, και η εκτίμηση διορθώνεται σε $\text{est}_i(t_k) = C_i$.
- Η P_i προσαρμόζει το ρολόι της με βάση τον μέσο όρο των εκτιμήσεων που κάνει για τα ρολόγια όλων των διεργασιών, $C_i = \sum_k \text{est}_i(t_k) / N$.

Ορθότητα

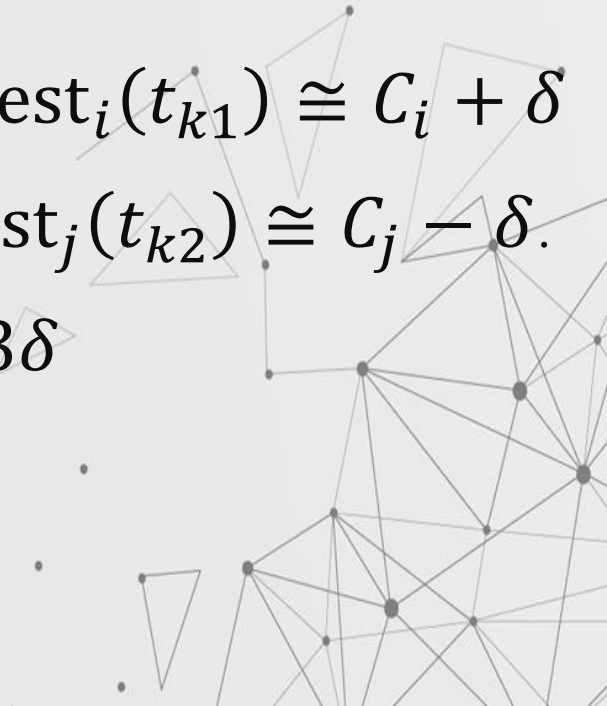
- Έστω ότι P_i και P_j λαμβάνουν t_{k1} και t_{k2} από την P_k .
- Αν η P_k δεν έχει βλάβη:

$$t_{k1} = t_{k2} \Rightarrow \text{est}_i(t_{k1}) = \text{est}_j(t_{k2})$$

- Αν η P_k έχει βλάβη, στην χειρότερη περίπτωση:

$$C_i - C_j \cong \delta$$

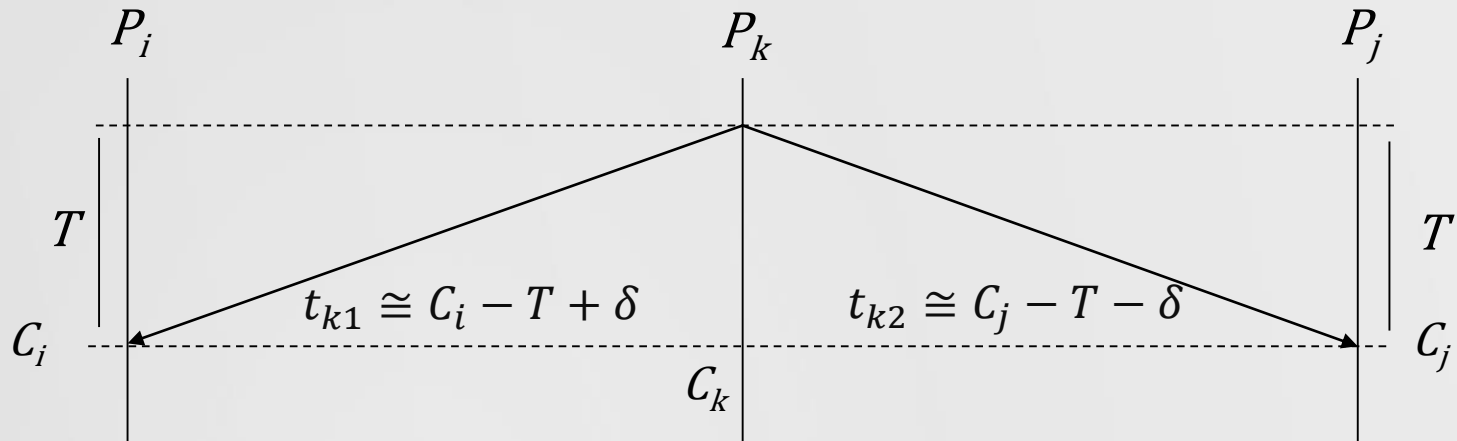
- t_{k1} η μεγαλύτερη δυνατή τιμή ώστε $\text{est}_i(t_{k1}) \cong C_i + \delta$
- t_{k2} η μικρότερη δυνατή τιμή ώστε $\text{est}_j(t_{k2}) \cong C_j - \delta$.

$$\text{est}_i(t_{k1}) - \text{est}_j(t_{k2}) \cong 3\delta$$


διεργασία με το πιο
γρήγορο ρολόι

βυζαντινή
διεργασία

διεργασία με το
πιο αργό ρολόι



$$C_i \cong X$$

$$\text{est}_i(t_{k1}) = t_{k1} + T \cong C_i + \delta$$

έστω

$$T_{\min} = T_{\max} = T$$

$$C_j \cong X - \delta$$

$$\text{est}_j(t_{k2}) = t_{k2} + T \cong C_j - \delta$$

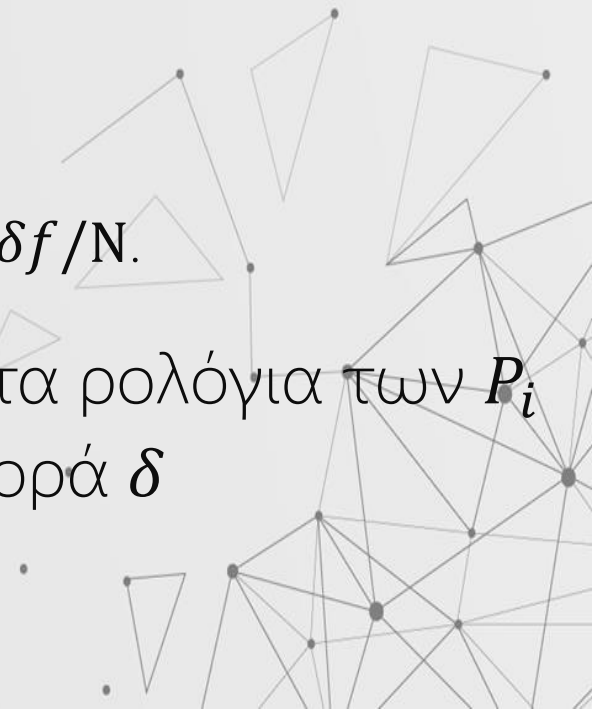
$$\begin{aligned} \text{est}_i(t_{k1}) - \text{est}_j(t_{k2}) &\cong (X + \delta) - (X - \delta - \delta) \\ &\cong \delta + \delta + \delta \\ &\cong 3\delta \end{aligned}$$

Ορθότητα (2)

- Για τις εκτιμήσεις των P_i και P_j για κάθε P_k , ισχύει:
 - P_k δεν έχει βλάβη $\Rightarrow |\text{est}_i(t_{k1}) - \text{est}_j(t_{k2})| = 0$.
 - P_k έχει βυζαντινή βλάβη $\Rightarrow |\text{est}_i(t_{k1}) - \text{est}_j(t_{k2})| \cong 3\delta$.
- Αφού υπάρχουν το πολύ f βυζαντινές διεργασίες:
 - $|\sum_k \text{est}_i(t_{k1}) - \sum_k \text{est}_j(t_{k2})| \leq 3\delta f$
- Επίσης:
 - $N \geq 3f + 1 \Rightarrow N > 3f \Rightarrow \delta N > 3\delta f \Rightarrow \delta > 3\delta f / N$.

Άρα, μετά την φάση του συγχρονισμού, τα ρολόγια των P_i και P_j είναι συγχρονισμένα, με άνω διαφορά δ

$$|c_i - c_j| < \delta$$





03

ΛΟΓΙΚΑ ΡΟΛΟΓΙΑ

Φυσικά ρολόγια και λογική σειρά

- Ο συγχρονισμός των φυσικών ρολογιών έχει κόστος
- Αν το ρολόι δεν είναι ακριβές ή οπισθοδρομήσει, για δύο τοπικά γεγονότα μπορεί $e; e' \wedge TS(e) \geq TS(e')$.
- Αν τα ρολόγια δεν είναι επακριβώς συγχρονισμένα, μπορεί $e \equiv snd(m) \wedge e' \equiv rcv(m) \wedge TS(e) \geq TS(e')$.
- Και στις 2 περιπτώσεις οι φυσικές χρονοσφραγίδες (timestamps - TS), αντιφάσκουν με την λογική σειρά των γεγονότων.
- Χρειαζόμαστε άλλα ρολόγια που να εγγυώνται την «χαλαρή συνθήκη ενός λογικά συνεπούς ρολογιού».
$$e \mapsto e' \wedge TS(e) < TS(e')$$

Λογικά ρολόγια Lamport

- Αρχικοποίηση $C_i = 0$.

- Συμβατικό τοπικό γεγονός $e_{i.x}$:

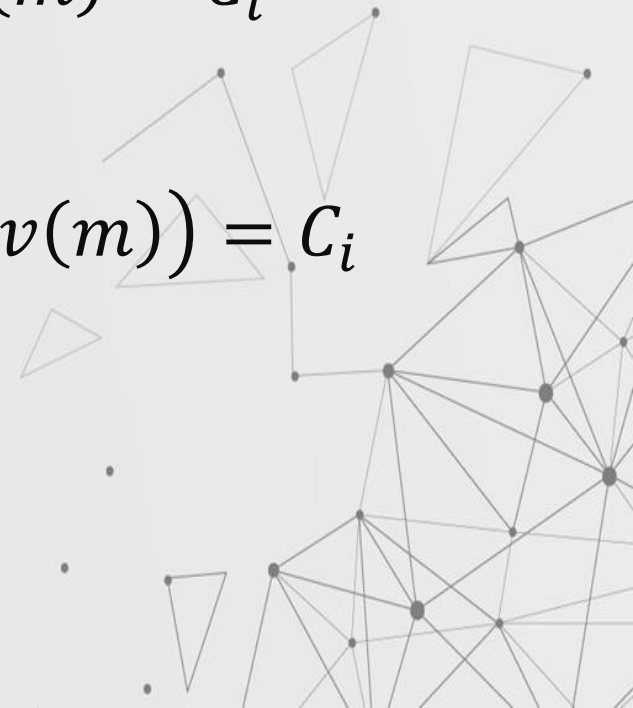
$$C_i = C_i + 1; TS(e_{i.x}) = C_i$$

- Αποστολή μηνύματος $snd(m)$:

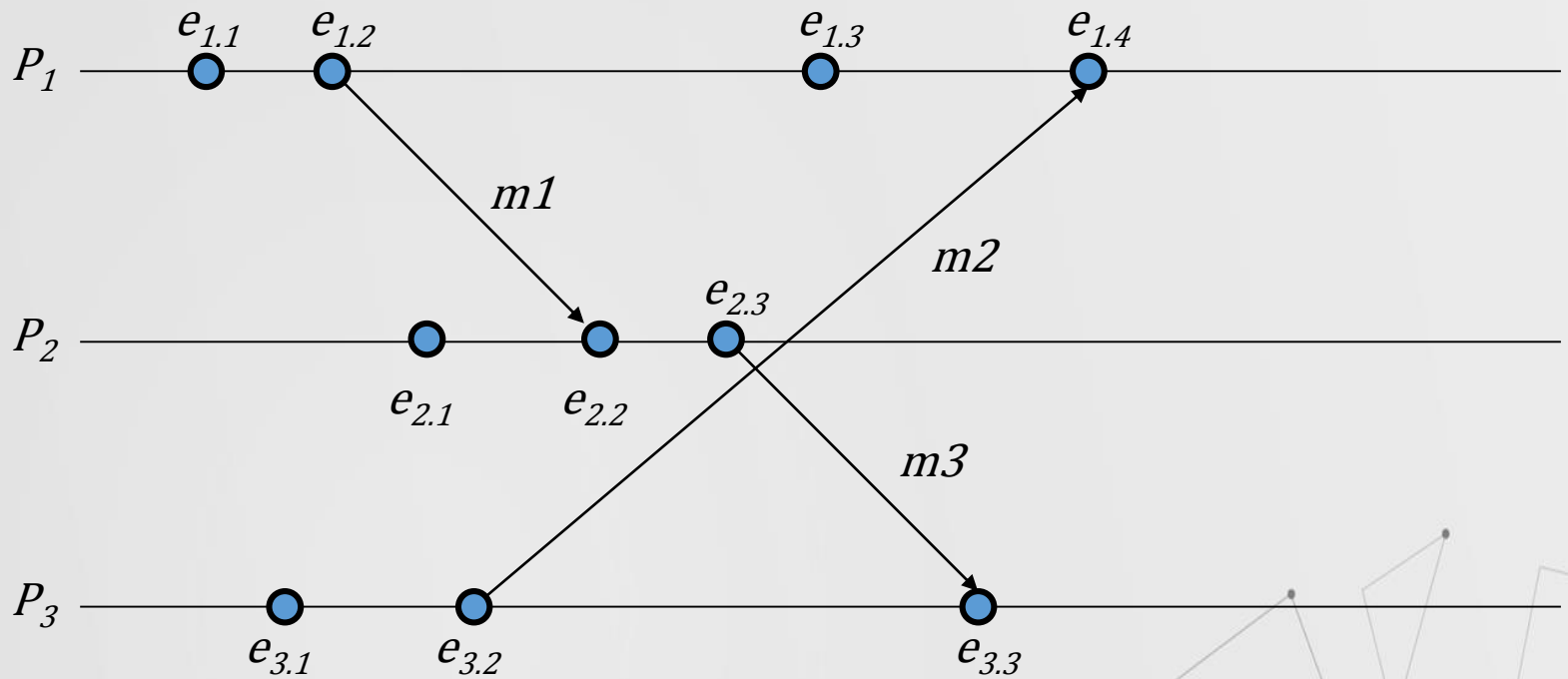
$$C_i = C_i + 1; TS(snd(m)), TS(m) = C_i$$

- Παραλαβή μηνύματος $rcv(m)$:

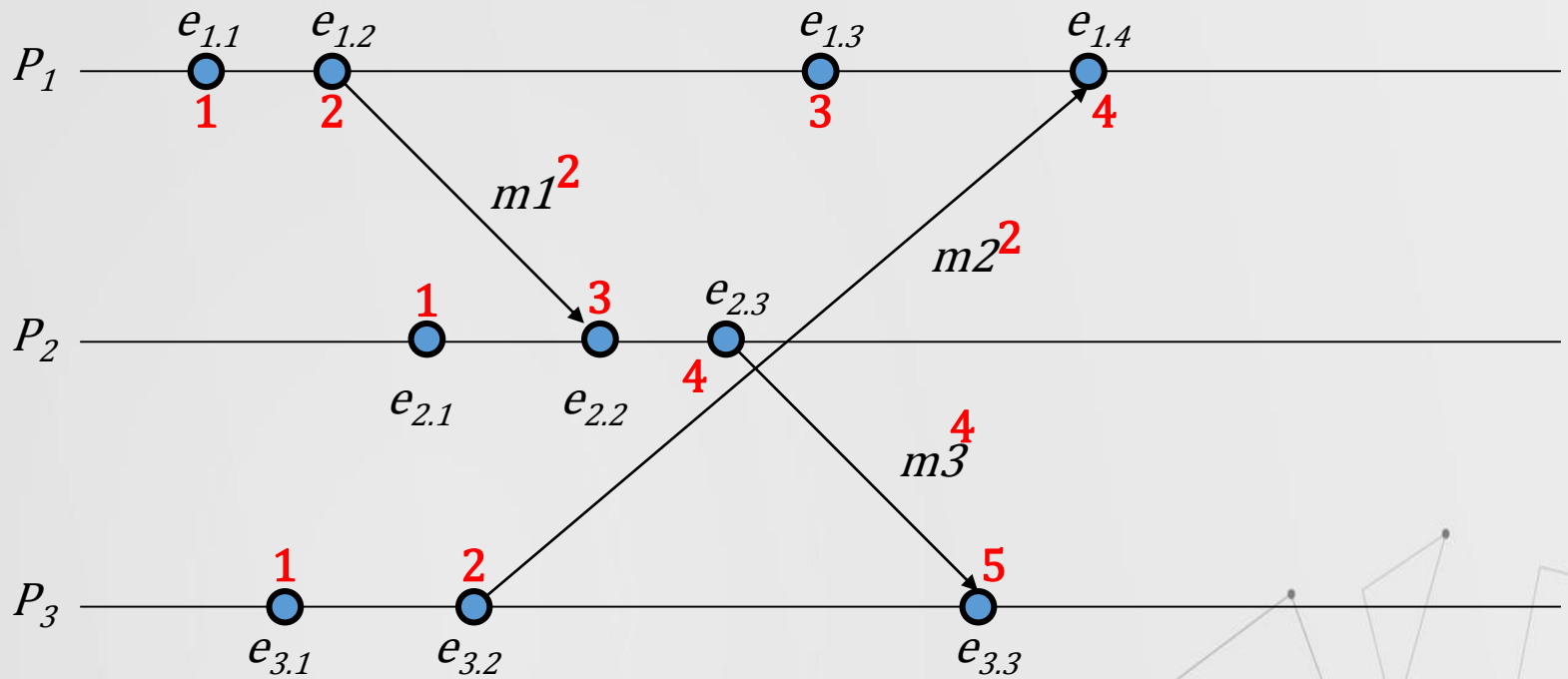
$$C_i = \max(C_i, TS(m) + 1); TS(rcv(m)) = C_i$$



Λογικά ρολόγια Lamport

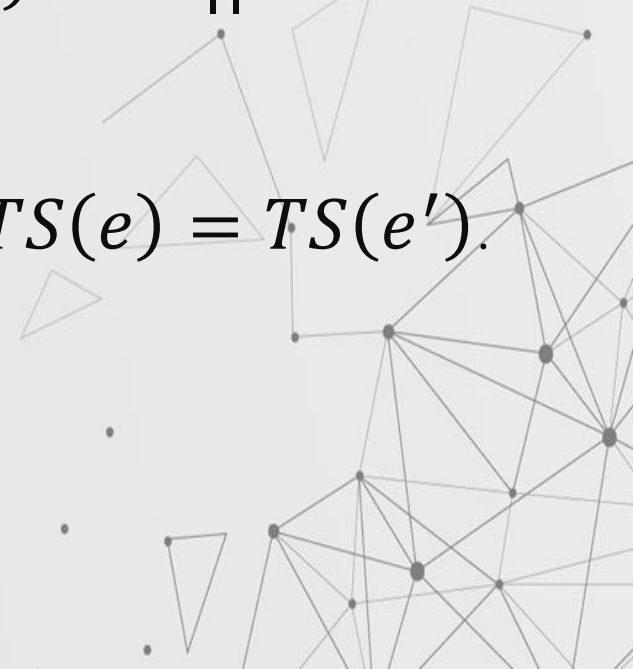


Λογικά ρολόγια Lamport

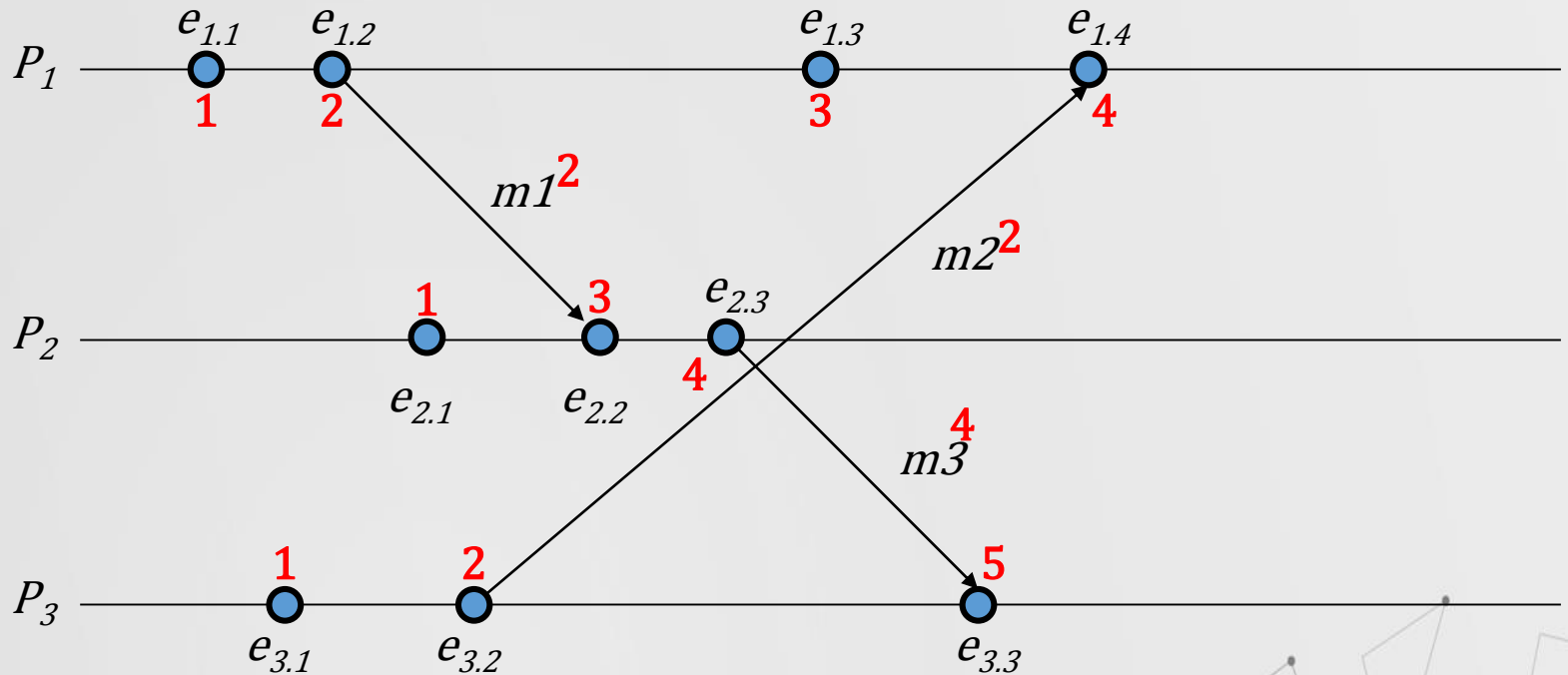


Ιδιότητες ρολογιών Lamport

- Λογική συνέπεια: $e \mapsto e' \Rightarrow TS(e) < TS(e')$.
- Δεν ισχύει το αντίστροφο: $TS(e) < TS(e') \not\Rightarrow e \mapsto e'$.
- Λογική ανεξαρτησία: $TS(e) = TS(e') \Rightarrow e \parallel e'$.
- Δεν ισχύει το αντίστροφο: $e \parallel e' \not\Rightarrow TS(e) = TS(e')$.



Ιδιότητες ρολογιών Lamport

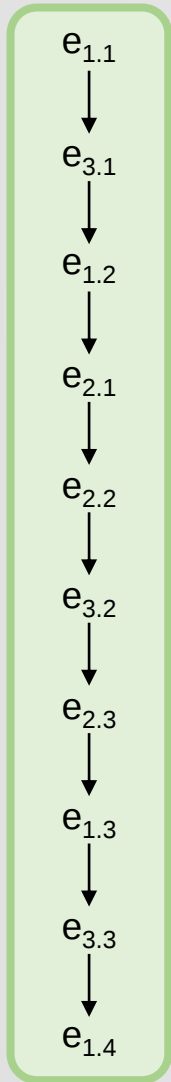


- $e_{1.1} \mapsto e_{3.3} \Rightarrow TS(e_{1.1}) < TS(e_{3.3})$.
- $TS(e_{3.2}) < TS(e_{2.3}) \not\Rightarrow e_{3.2} \mapsto e_{2.3}$.
- $TS(e_{3.1}) = TS(e_{1.1}) \Rightarrow e_{3.1} \parallel e_{1.1}$.
- $e_{3.2} \parallel e_{1.3} \not\Rightarrow TS(e_{3.2}) = TS(e_{1.3})$.

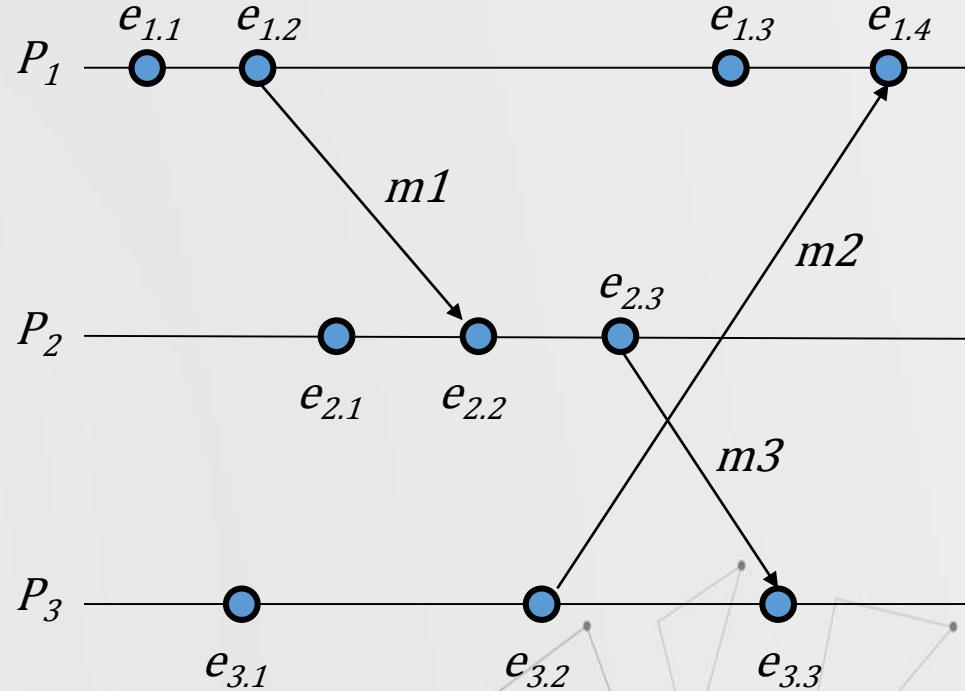
Αδυναμία ρολογιών Lamport (1)

- Τα ρολόγια Lamport μπορεί να χρησιμοποιηθούν για να δώσουμε timestamps που είναι συνεπή με την λογική σειρά, αλλά δεν την καταγράφουν ακριβώς.
 - παρόμοια αδυναμία όπως των φυσικών ρολογιών.
 - είναι όμως σαφώς πιο κατάλληλα για την καταγραφή της λογικής σειράς από τα φυσικά ρολόγια.
- Δεν καταγράφεται ακριβώς η λογική ανεξαρτησία.
- Ένα μονοδιάστατο μέγεθος (όπως η τιμή του ενός φυσικού ρολογιού ή ρολογιού Lamport) δεν μπορεί να περιγράψει την σχέση λογικής εξάρτησης (στην γενικότερη περίπτωση δεν είναι μια απλή αλυσίδα).

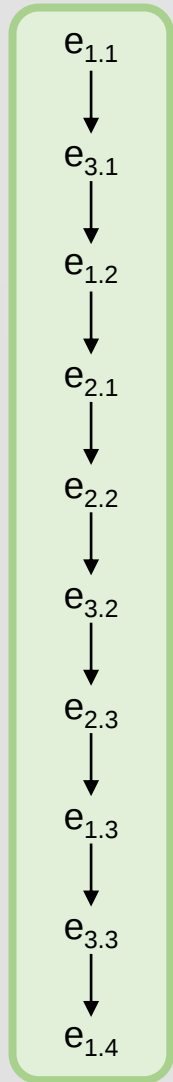
Αδυναμία ρολογιών Lamport (2)



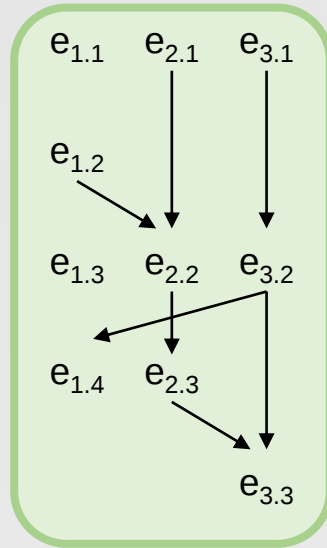
time order



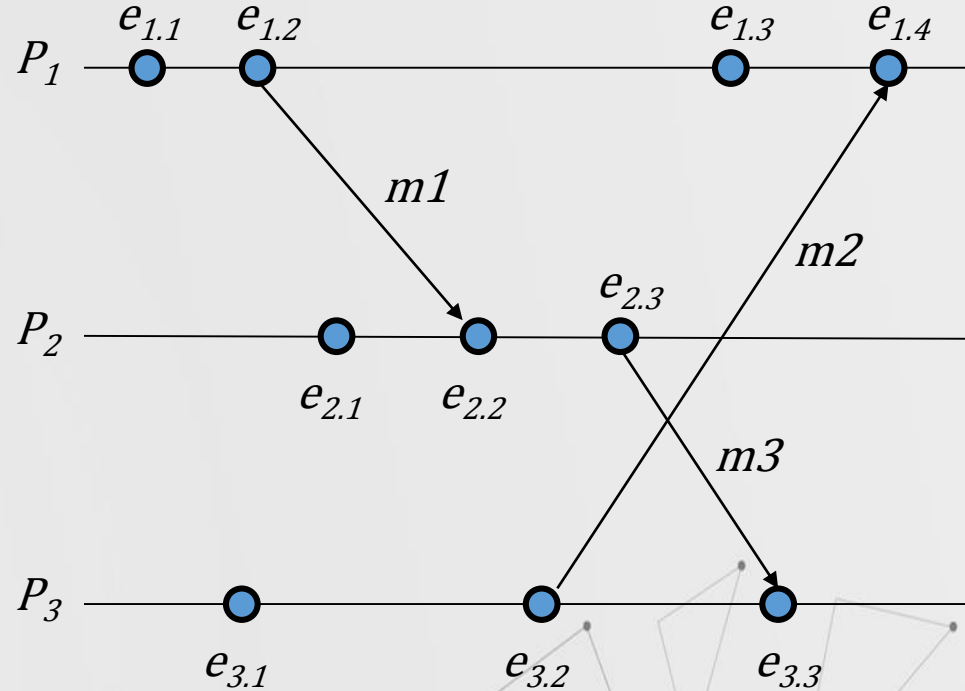
Αδυναμία ρολογιών Lamport (3)



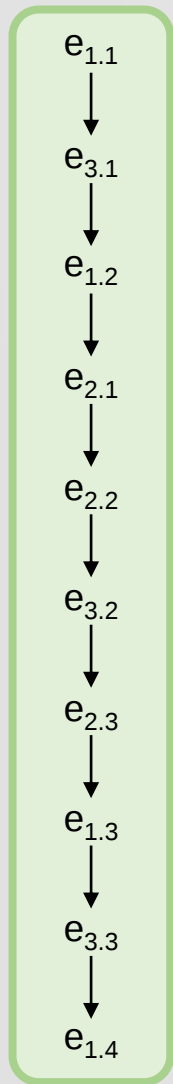
time order



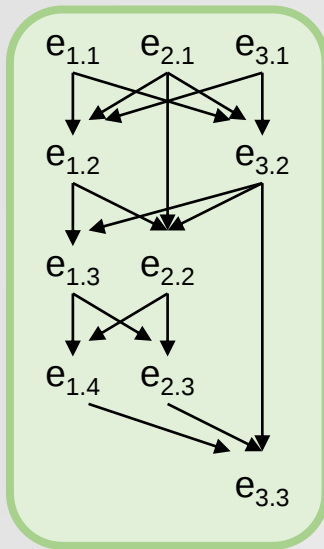
logical order



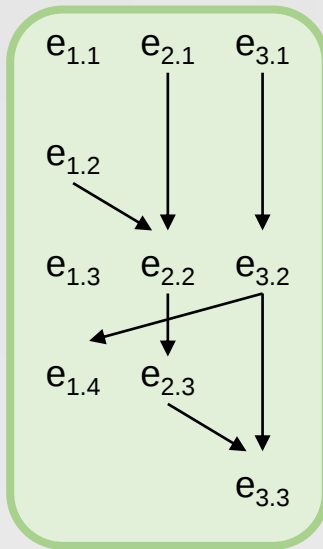
Αδυναμία ρολογιών Lamport (4)



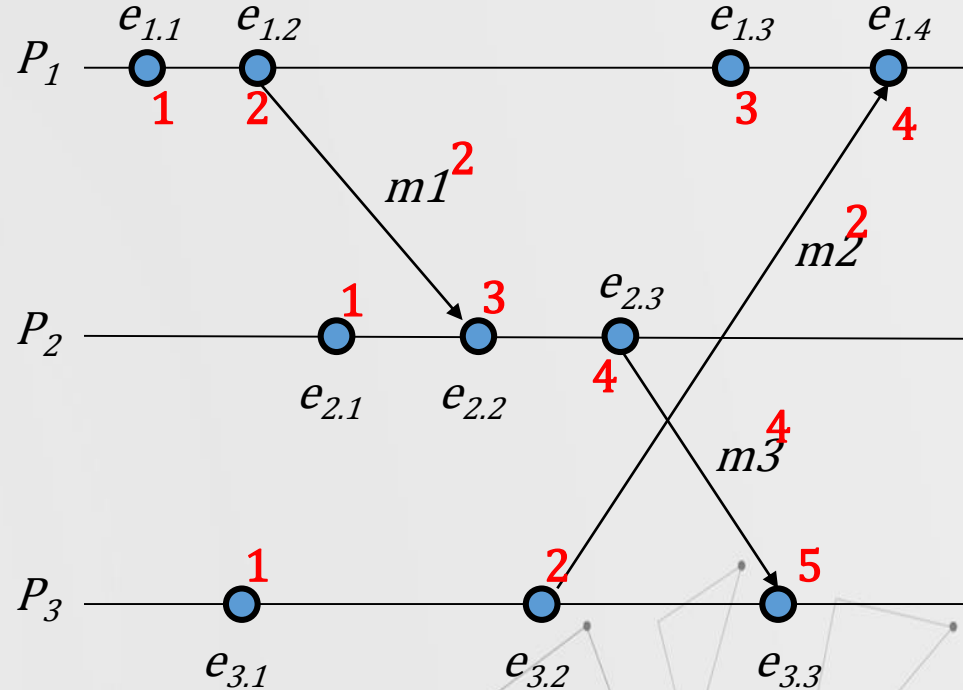
time order



Lamport order

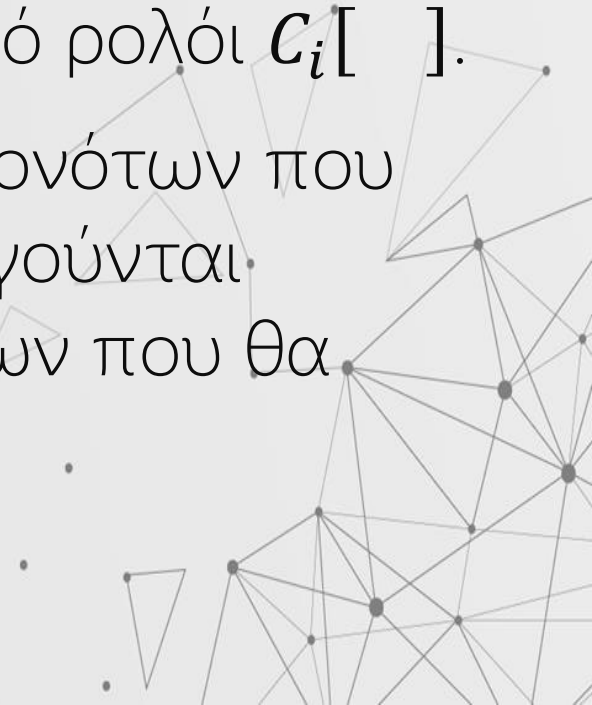


logical order



Πέρα από τα μονοδιάστατα ρολόγια

- Επέκταση του μονοδιάστατου λογικού ρολογιού.
- Για να εντοπιστεί η σχέση λογικής εξάρτησης, αρκεί να γνωρίζουμε το πλήθος των λογικά προηγούμενων γεγονότων σε κάθε μια από τις διεργασίες.
- Χρησιμοποιούμε διανυσματικό λογικό ρολόι $C_i[\quad]$.
- Όπου $C_i[k]$ είναι το πλήθος των γεγονότων που έχουν λάβει χώρα στην P_k και προηγούνται αιτιολογικά των επόμενων γεγονότων που θα λάβουν χώρα στην P_i .



Διανυσματικά ρολόγια

- Αρχικοποίηση $C_i[k] = 0$.

- Συμβατικό τοπικό γεγονός $e_{i.x}$:

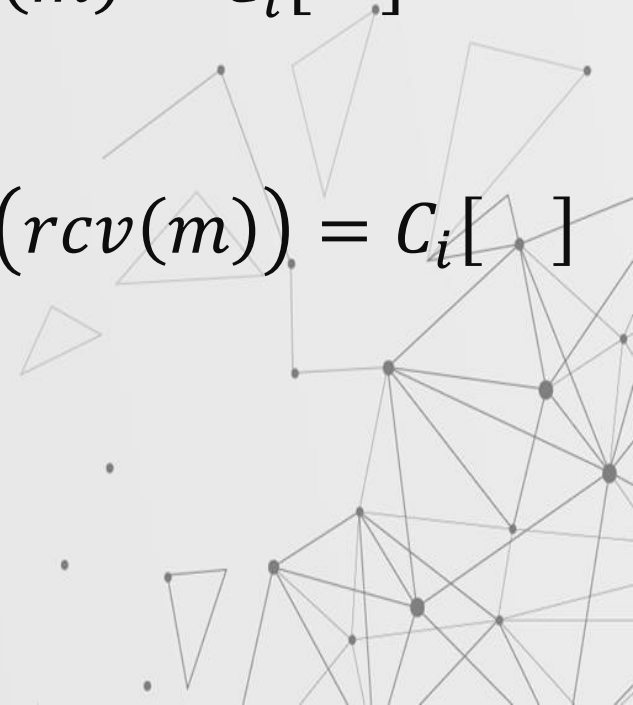
$$C_i[i] = C_i[i] + 1; TS(e_{i.x}) = C_i[i]$$

- Αποστολή μηνύματος $snd(m)$:

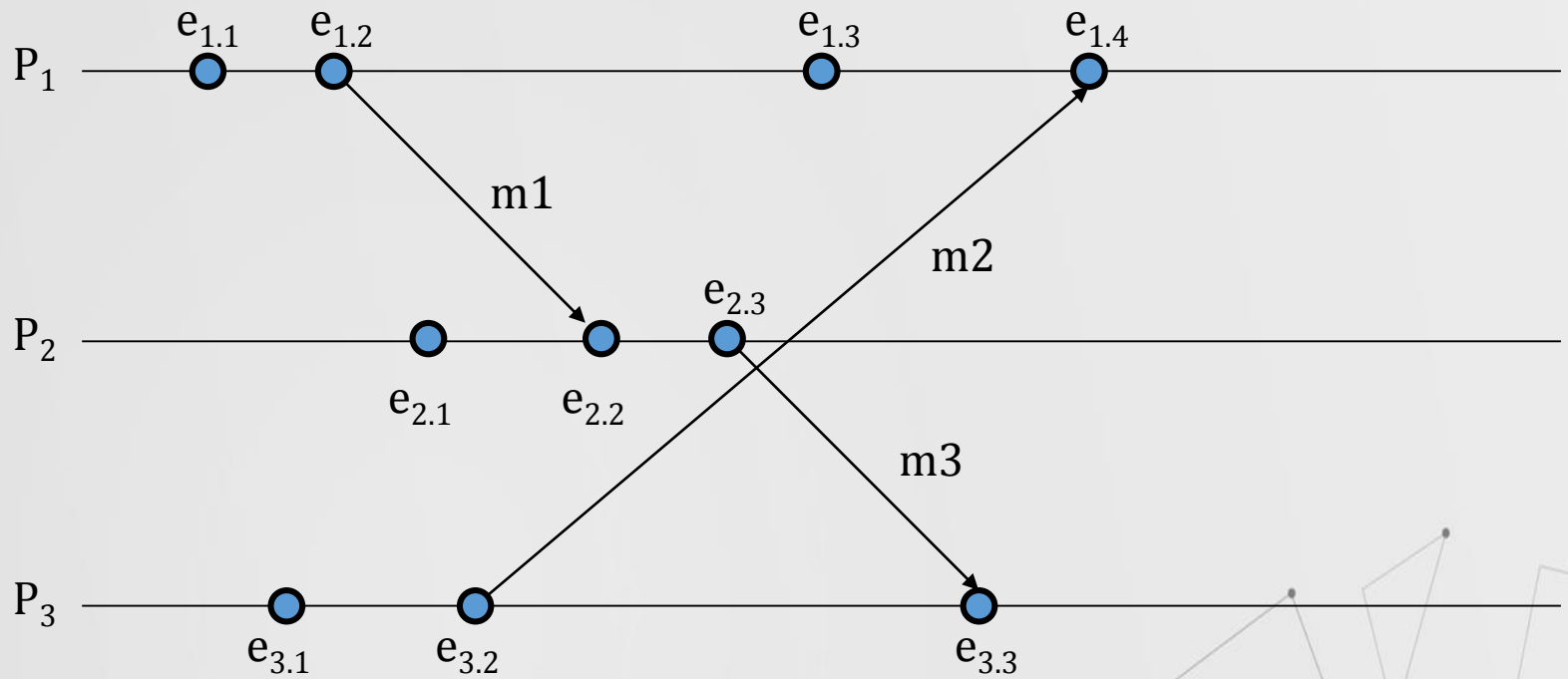
$$C_i[i] = C_i[i] + 1; TS(snd(m)), TS(m) = C_i[i]$$

- Παραλαβή μηνύματος $rcv(m)$:

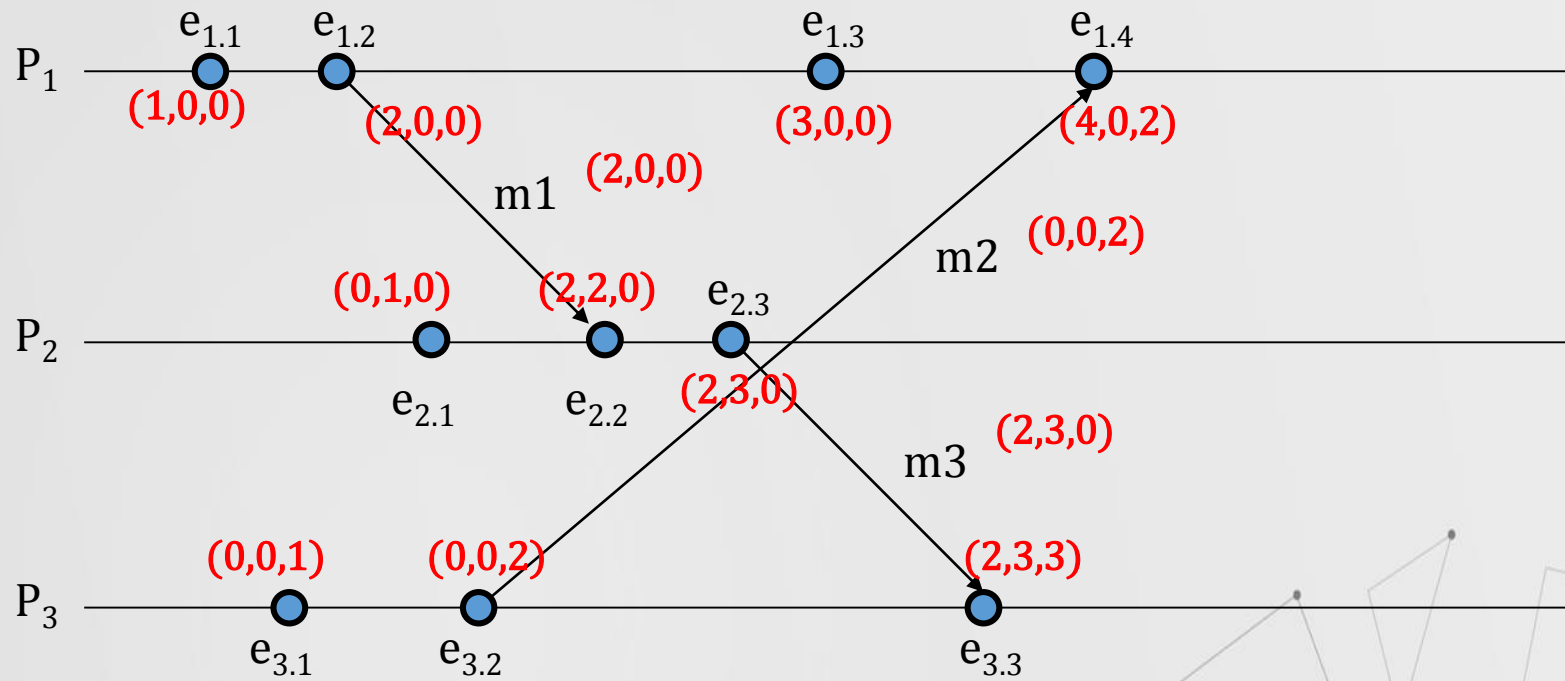
$$C_i[k] = \max(C_i[k], TS(m)[k]) + 1; TS(rcv(m)) = C_i[k]$$



Διανυσματικά ρολόγια



Διανυσματικά ρολόγια



Σύγκριση διανυσματικών χρονοσφραγίδων

- Ισότητα:

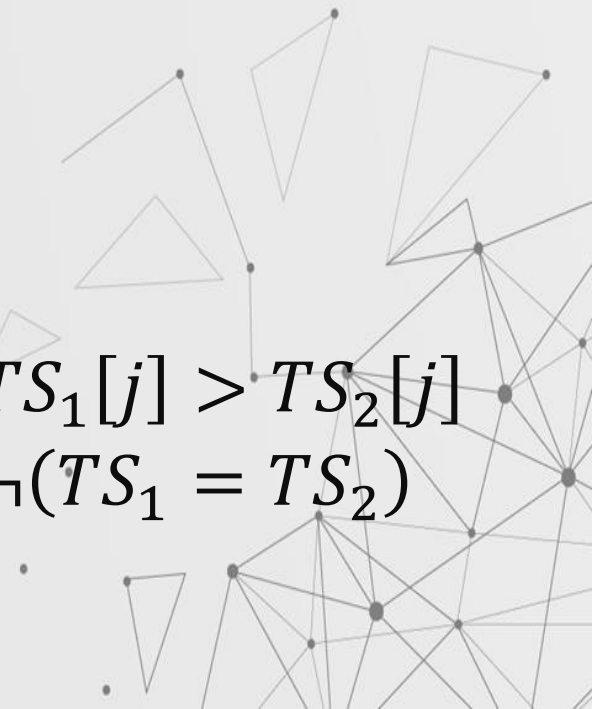
$$TS_1 = TS_2 \equiv TS_1[k] = TS_2[k], \forall k$$

- Ανισότητα:

$$TS_1 < TS_2 \equiv \exists i : TS_1[i] < TS_2[i], \\ \wedge \forall j \neq i : TS_1[j] \leq TS_2[j]$$

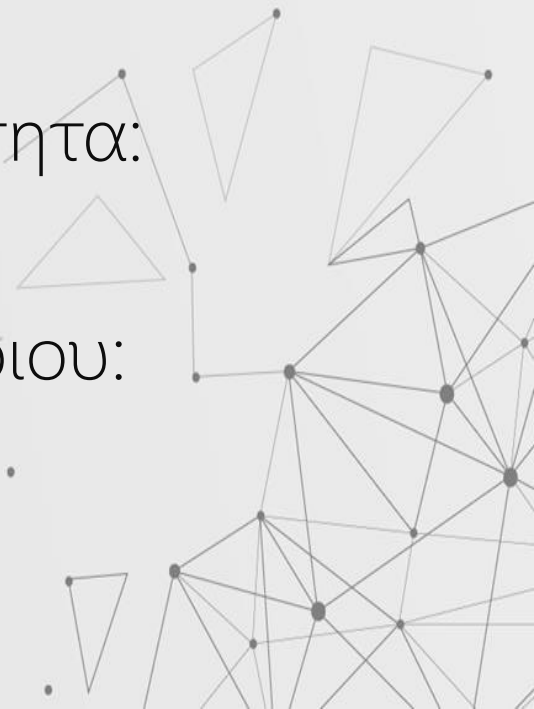
- Μη συγκρίσιμες τιμές:

$$TS_1 \text{ ΜΣ } TS_2 \equiv \exists i, j : TS_1[i] < TS_2[i] \wedge TS_1[j] > TS_2[j] \\ \equiv \neg(TS_1 < TS_2) \wedge \neg(TS_1 > TS_2) \wedge \neg(TS_1 = TS_2)$$

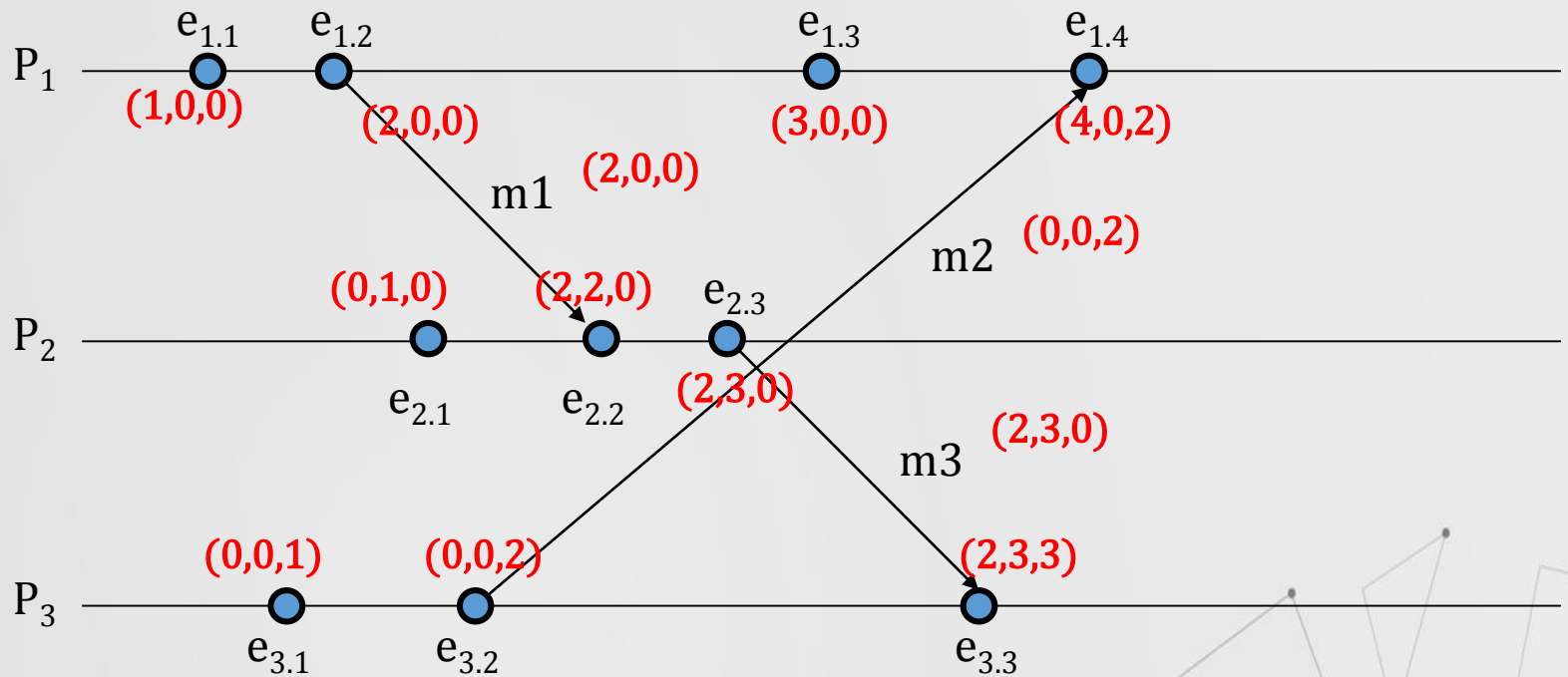


Αυστηρή συνθήκη λογικά συνεπούς ρολογιού

- Με τις διανυσματικές χρονοσφραγίδες καταγράφεται επακριβώς η σχέση λογικής εξάρτησης
$$e \mapsto e' \Leftrightarrow TS(e) < TS(e')$$
- Με βάση τις χρονοσφραγίδες δύο γεγονότων, και χωρίς κάποια επιπλέον πληροφορία, μπορούμε να συμπεράνουμε κατά πόσο
- (α) δύο γεγονότα είναι λογικά ανεξάρτητα:
$$TS(e) \not\leq TS(e') \Rightarrow e \parallel e'$$
- (β) ποιο γεγονός προηγείται λογικά ποιου:
$$TS(e) < TS(e') \Rightarrow e \mapsto e' .$$
- Επίσης: $TS(e) = TS(e') \Rightarrow e \equiv e'$



Παράδειγμα



$$TS(e_{1.1}) < TS(e_{3.3}) \Rightarrow e_{1.1} \mapsto e_{3.3}$$

$$TS(e_{3.2}) \text{ M}\Sigma TS(e_{2.3}) \Rightarrow e_{3.2} || e_{2.3}$$