10 Κατανεμημένα Συστήματα

Ομαδική Επικοινωνία (Group Communication)

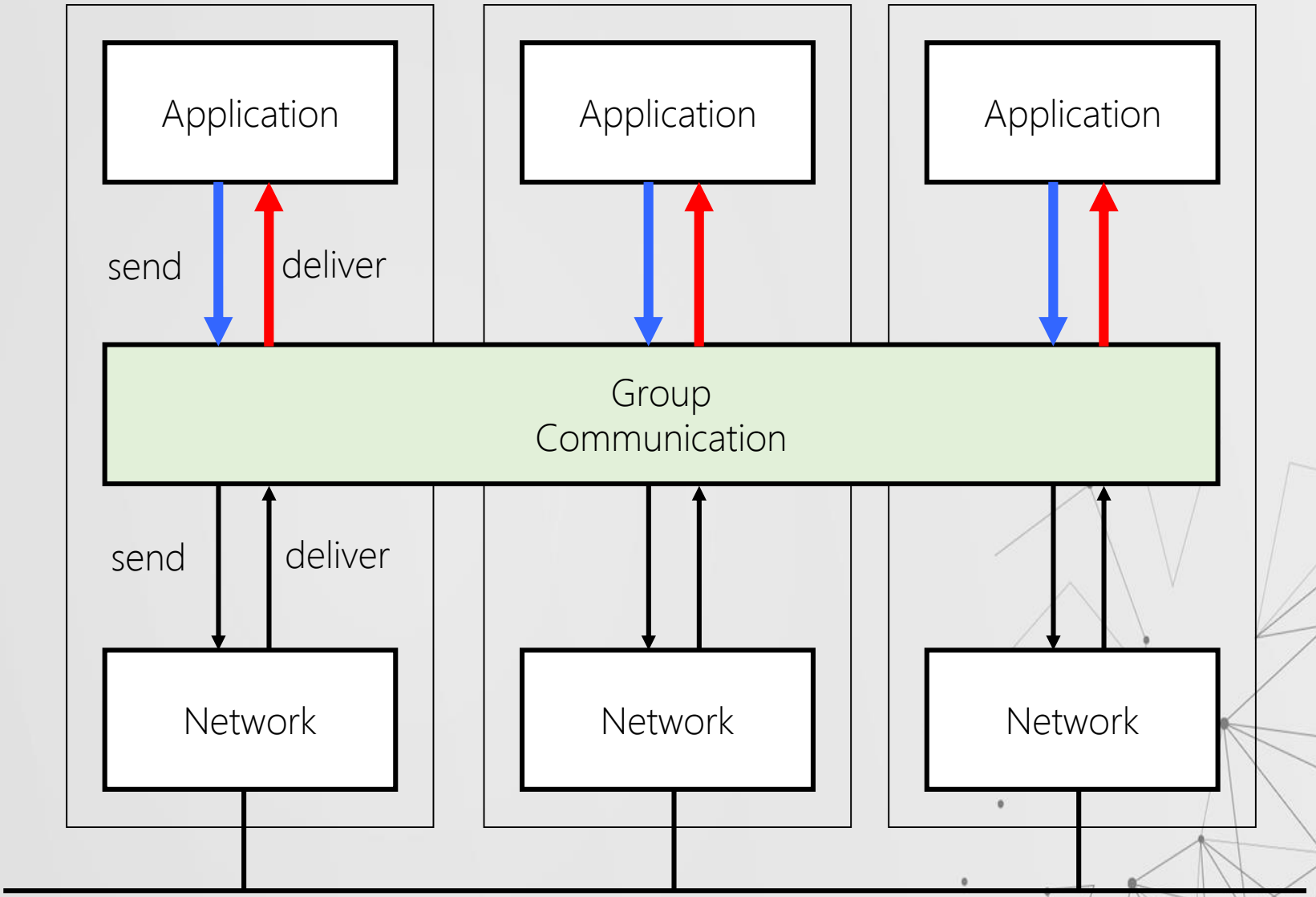
Ομαδική επικοινωνία

- Σαφής διαχωρισμός ανάμεσα στις διεργασίες που είναι και σε αυτές που δεν είναι μέλη της ομάδας.
- Ανοιχτή ομαδική επικοινωνία: ο αποστολέας δεν χρειάζεται να είναι απαραίτητα μέλος της ομάδας.
- Κλειστή ομαδική επικοινωνία: ο αποστολέας πρέπει να είναι απαραίτητα μέλος της ομάδας.
- Θα εξετάσουμε την κλειστή ομαδική επικοινωνία, με τη μορφή μιας μεσοστρωματικής υπηρεσίας, εστιάζοντας στα πιο βασικά θέματα
  1. Αξιοπιστία μετάδοσης μηνυμάτων
  2. Διαχείριση σύνθεσης ομάδας
  3. Σειρά παράδοσης μηνυμάτων

Μοντέλο/υποθέσεις εργασίας

- Υπηρεσία δικτύου/μεταφοράς
  - **send**: αποστολή μηνύματος (top-down).
  - **deliver**: παράδοση μηνύματος (bottom-up).
- Υπηρεσία ομαδικής επικοινωνίας.
  - **send**: αποστολή ομαδικού μηνύματος.
  - **deliver**: παράδοση ομαδικού μηνύματος.
- Βλάβες fail-stop: η διεργασία σταματά να λειτουργεί και (κάπως) ενημερώνονται όλες οι υπόλοιπες.
- Το δίκτυο είναι αξιόπιστο: δεν χάνει μηνύματα, ούτε δημιουργεί από μόνο του διπλότυπα.

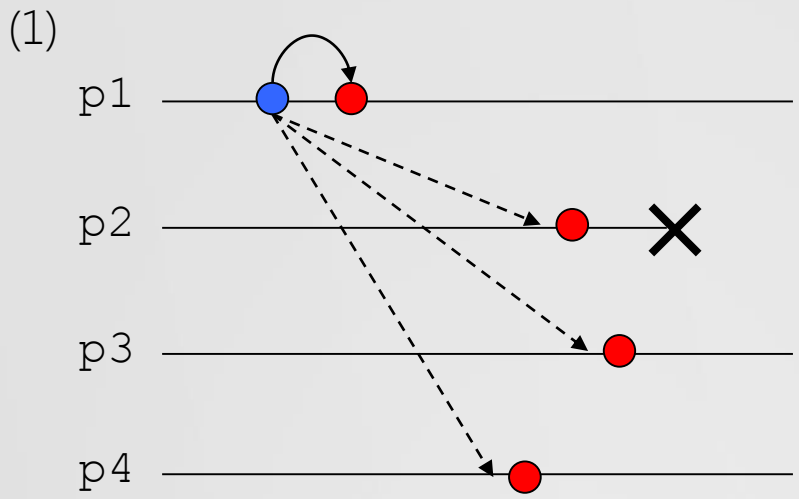
Μοντέλο/υποθέσεις εργασίας

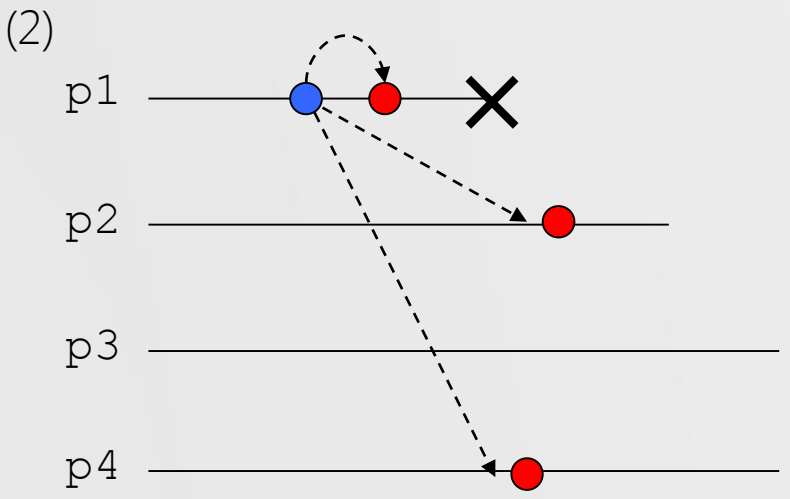


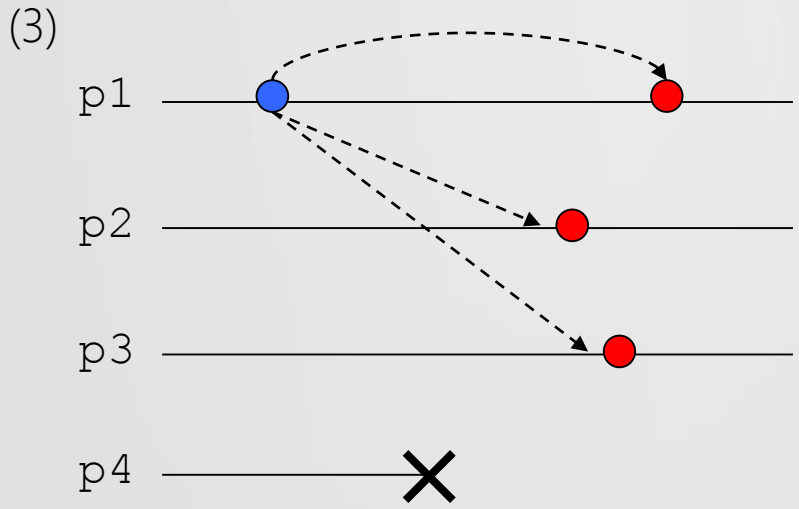
Basic multicast (BM)

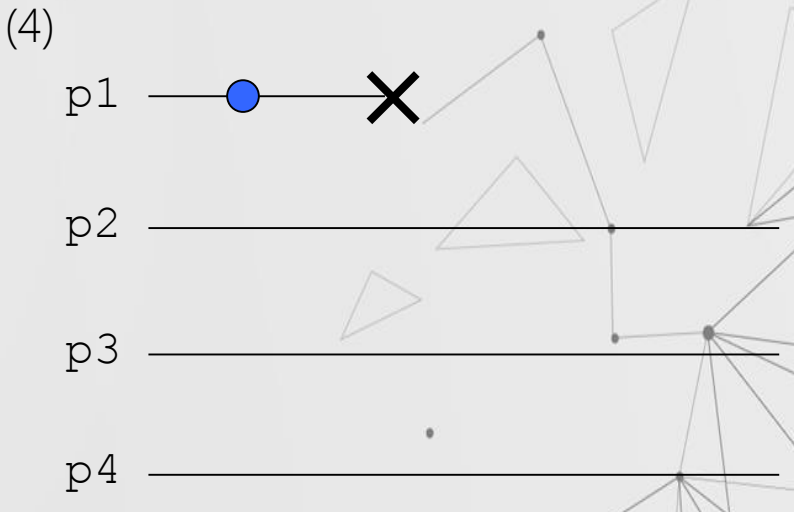
- Κάθε ομαδικό μήνυμα παραδίδεται το πολύ μια φορά σε κάθε διεργασία/κόμβο που είναι μέλος της ομάδας.
- Αν ο αποστολέας δεν παρουσιάσει βλάβη, το μήνυμα θα παραδοθεί σε όλα τα μέλη (που δεν έχουν βλάβη).
- Αν ο αποστολέας παρουσιάσει βλάβη (κατά την διαδικασία της αποστολής), δεν δίνεται καμία εγγύηση αναφορικά με τη παράδοση του μηνύματος στα μέλη.
  - το μήνυμα μπορεί τελικά να παραδοθεί σε όλα τα μέλη, σε κάποια μέλη ή σε κανένα μέλος της ομάδας.

Basic multicast – Παράδειγμα

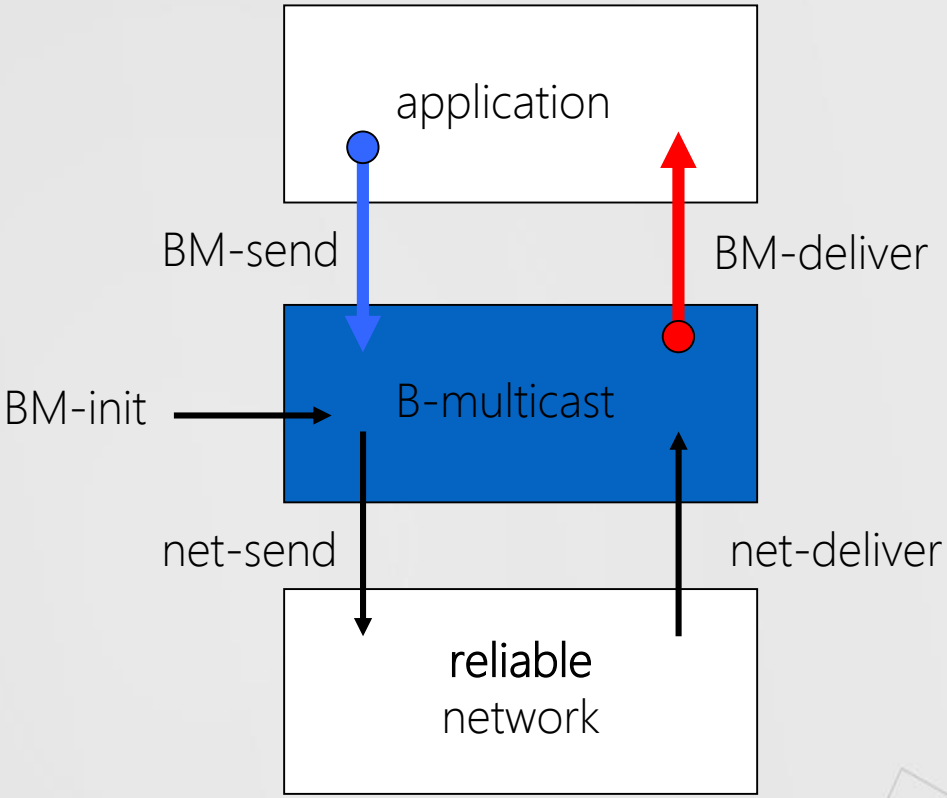








Basic multicast – Παράδειγμα



Παρατηρήσεις

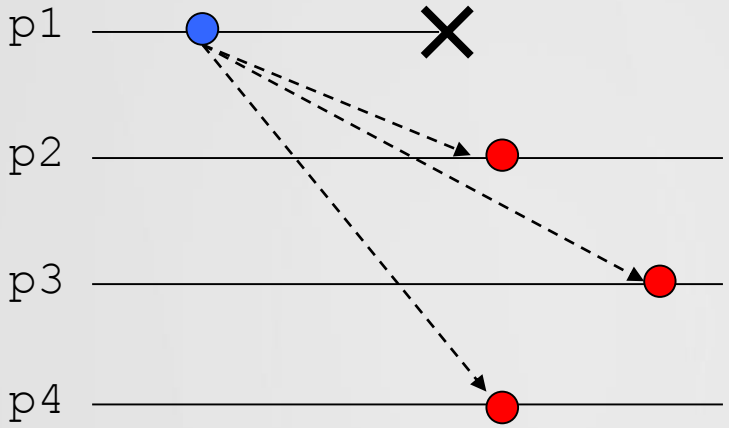
- Η επανειλημμένη αποστολή προφανώς αποτελεί μη βέλτιστη λύση για μια πραγματική υλοποίηση.
- Μπορεί να γίνουν διάφορες βελτιστοποιήσεις.
  - βλέπε επικοινωνία 1→N.
- Αν το δίκτυο δεν είναι αξιόπιστο, το πρωτόκολλο πρέπει να επεκταθεί κατάλληλα
  - επιβεβαιώσεις, επανεκπομπές, αναγνώριση διπλοτύπων …

Reliable (atomic) multicast (RM)

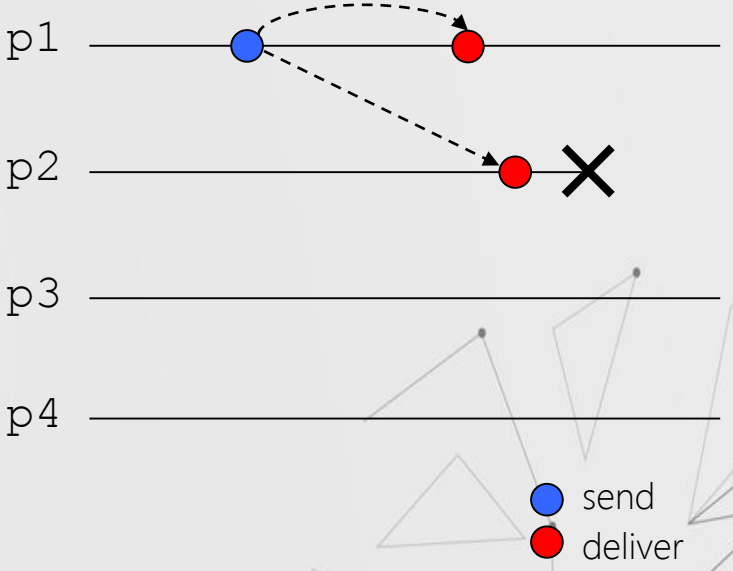
- Αν υπάρχει τουλάχιστον ένα μέλος της ομάδας που δεν παρουσιάζει βλάβη και έλαβε το μήνυμα, το μήνυμα θα παραδοθεί τελικά σε όλα τα μέλη της ομάδας (που δεν παρουσιάζουν βλάβη).
- Η προώθηση μηνυμάτων γίνεται κοινή ευθύνη όλων, όχι μόνο του αρχικού αποστολέα.
- Απλή υλοποίηση: κάθε μέλος στο οποίο φτάνει ένα (νέο) ομαδικό μήνυμα, το ξαναστέλνει σε όλους.

Reliable (atomic) multicast (RM)

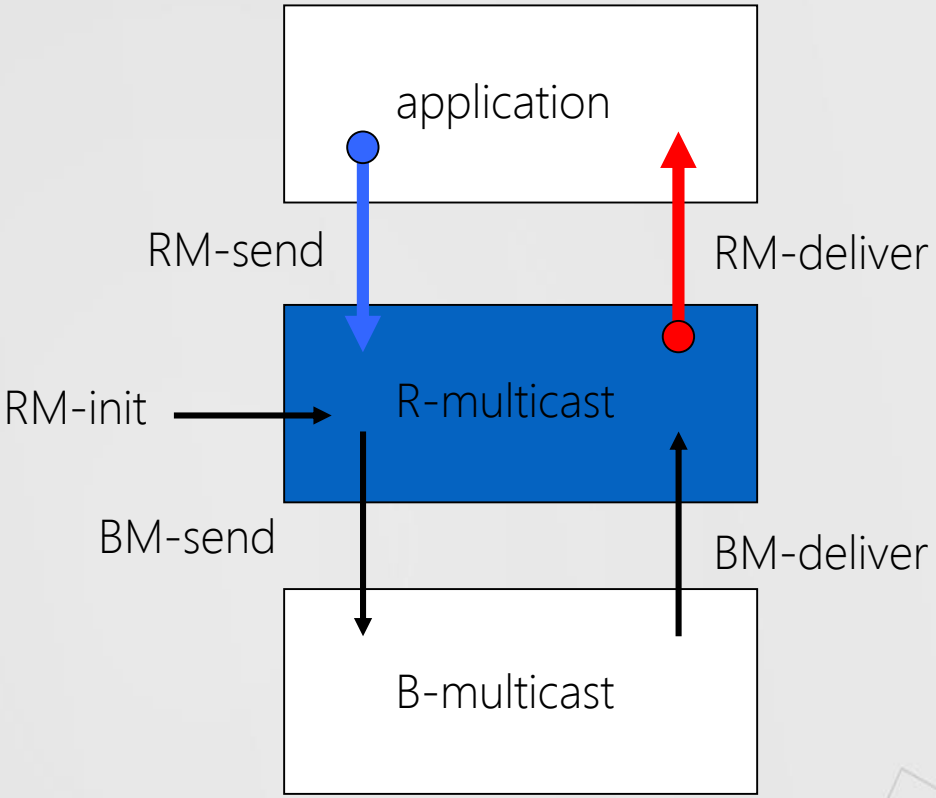
(1) reliable multicast



(2) NOT reliable multicast



Reliable (atomic) multicast (RM)



Παρατηρήσεις

- Αφού ένα μήνυμα ξαναστέλνεται σε πιο ψηλό επίπεδο, πρέπει να υπάρχει αντίστοιχος εντοπισμός διπλοτύπων.
- Αντί κάθε νέο μήνυμα να ξαναστέλνεται στα «τυφλά» σε όλα τα μέλη της ομάδας, μπορεί να στέλνεται μόνο στα μέλη που (πιθανώς) δεν το έχουν λάβει (ακόμα).
- Απαιτείται κατάλληλη επέκταση.

Αντιμετώπιση βλαβών

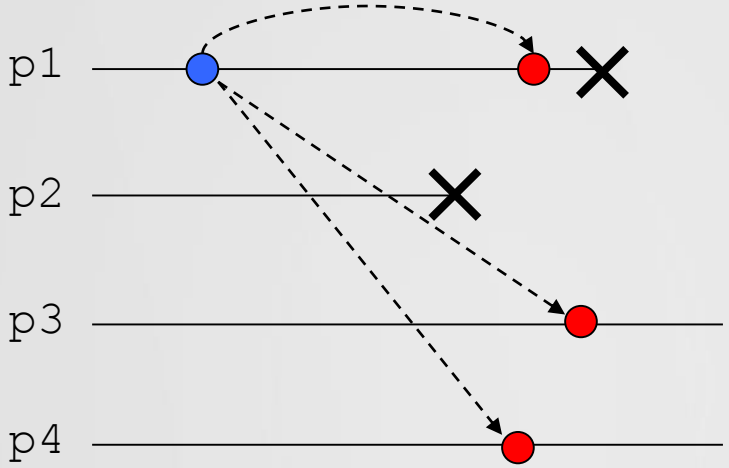
- Υποθέτουμε fail-stop, δηλαδή υπάρχει μηχανισμός εντοπισμού βλαβών και ειδοποίησης διεργασιών.
- Τα προηγούμενα πρωτόκολλα πρέπει να επεκταθούν για να γίνει κατάλληλος χειρισμός της βλάβης ενός μέλους.
- Η διεργασία πρέπει να αφαιρείται από την ομάδα, και να μην λαμβάνεται πλέον υπ' όψη για την αποστολή μηνυμάτων και παραλαβή επιβεβαιώσεων.
- Αν εξακολουθούν να εκκρεμούν επιβεβαιώσεις από την διεργασία που παρουσίασε βλάβη, τότε αυτές απλά αγνοούνται.

Uniform reliable multicast

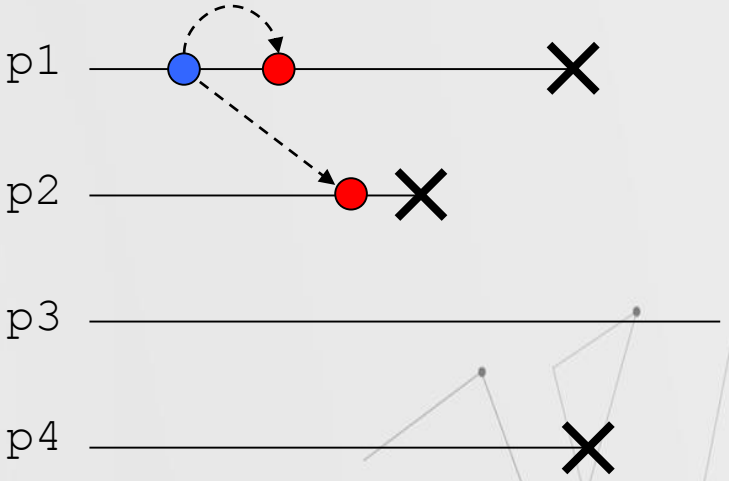
- Ένα μήνυμα παραδίδεται στην εφαρμογή μόνο όταν πλέον έχει φτάσει, σε επίπεδο υπηρεσίας, σε όλα τα (υφιστάμενα) μέλη της ομάδας.
- Η διαφορά σε σχέση με το απλό RM είναι πως ένα μέλος της ομάδας που δεν θα παρουσιάζει βλάβη δεν μπορεί να «χάσει» μηνύματα που έχουν ήδη παραδοθεί σε μέλη της ομάδας που στην συνέχεια ίσως παρουσιάσουν βλάβη.
- Αυξημένη «λογική συνέπεια» σε περίπτωση βλάβης.
- Η χρησιμότητά της εξαρτάται από την εφαρμογή.

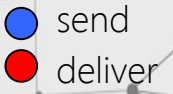
Uniform reliable multicast

(1) uniform reliable multicast



(2) NOT uniform reliable multicast

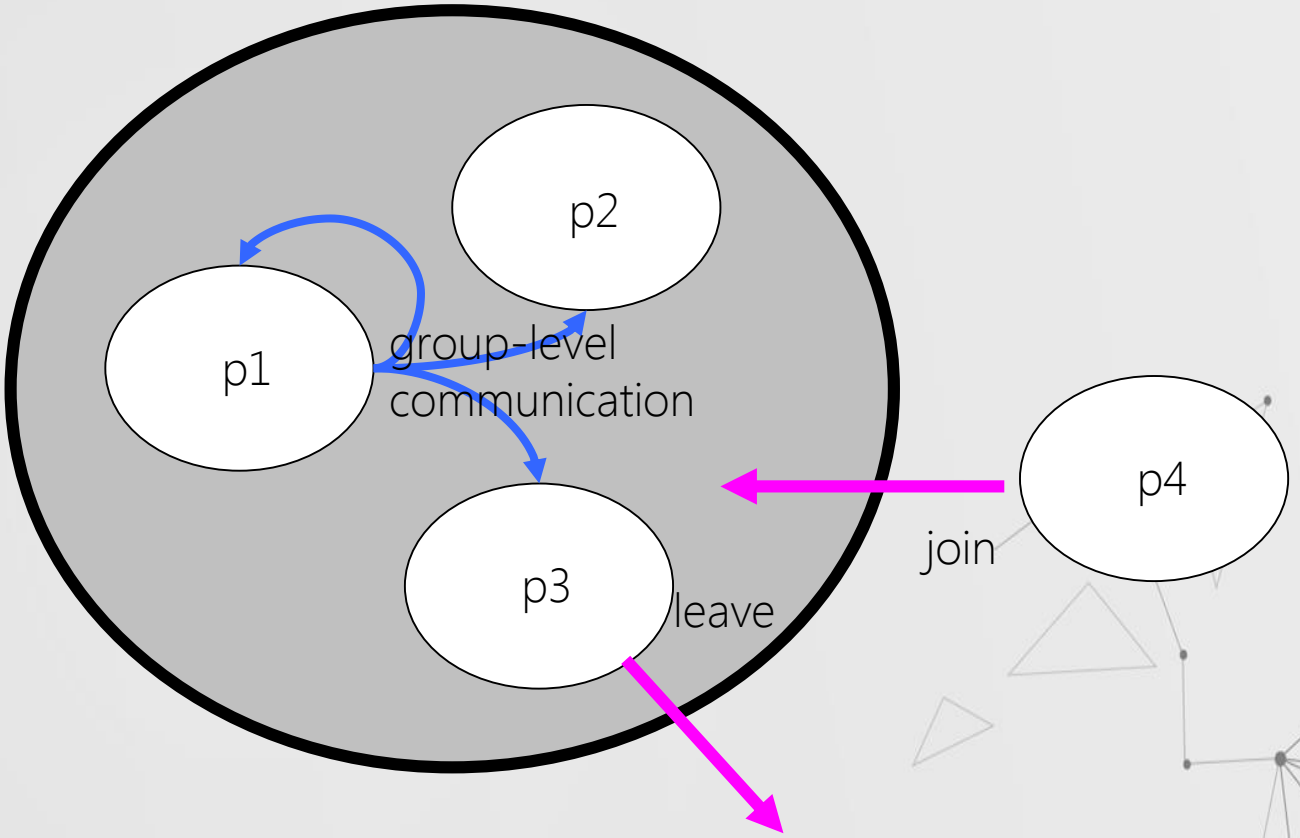




Διαχείριση σύνθεσης ομάδας

- Τι γίνεται αν η σύνθεση της ομάδας αλλάζει;
- Απομάκρυνση μέλους (λόγω βλάβης).
- Αποχώρηση μέλους (εθελοντικά).
- Προσθήκη μέλους.
- Απαιτείται μηχανισμός διαχείρισης (σύνθεσης) ομάδας.
- Ανανέωση σύνθεσης σε επίπεδο εφαρμογής.
- Συνεπής παράδοση μηνυμάτων.

Διαχείριση σύνθεσης ομάδας



Εικόνα τρέχουσας σύνθεσης

- Ιδανικό: όλα τα μέλη έχουν ανά πάσα στιγμή την ίδια εικόνα για την σύνθεση της ομάδας.
- $\forall P_1, P_2 \in g : \mathbf{view}(P_1, t) = \mathbf{view}(P_2, t)$
- Σύγχρονο σύστημα: μπορεί να επιτευχθεί μερικώς, με άνω όριο στην καθυστέρηση/απόκλιση ενημέρωσης για αλλαγές σύνθεσης της ομάδας ανάμεσα στα μέλη.
- Ασύγχρονο σύστημα: πρακτικά ανέφικτο.
- Εφικτό: ενημέρωση των μελών με κοινή σειρά και επιβεβαίωση.

Μηνύματα (αλλαγής) σύνθεσης

- Ο μηχανισμός διαχείρισης σύνθεσης ενημερώνει τα μέλη στέλνοντας τους την τρέχουσα σύνθεση της ομάδας (view) κάθε φορά που αυτή αλλάζει.
- $V = \{P_1, P_2, \dots, P_N\} \rightarrow P_1 \in g \cap P_2 \in g \cap \cdots \cap P_N \in g$
- Ενημέρωση σύνθεσης V στην P την χρονική στιγμή t.
- $\mathbf{update}(P, V, t)$
- Αποστολή μηνύματος και παραλαβή μηνύματος στην P στο πλαίσιο της τρέχουσας σύνθεσης V.
- $\mathbf{send}(m)@P|V$, $\mathbf{recv}(m)@P|V$.

Ιδιότητες ενημέρωσης σύνθεσης

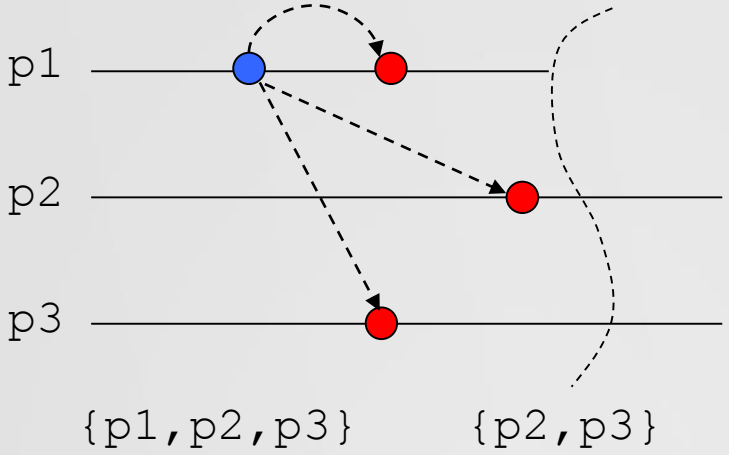
- Πρόοδος ενημέρωσης: αν P_1 λάβει την σύνθεση V, κάθε άλλο μέλος P_2 της ομάδας (εφόσον υφίσταται) έχει/θα λάβει την σύνθεση V:
- $\mathbf{update}(P_1, V, t) \rightarrow \forall P_2 \in g : \mathbf{update}(P_2, V, t')$
- Συνέπεια ενημέρωσης: τα μέλη ενημερώνονται για τις αλλαγές σύνθεσης της ομάδας με την ίδια σειρά:
- $\mathbf{update}(P_1, V_1, t_1) \cap \mathbf{update}(P_2, V_2, t_2) \cap t_1 < t_2 \rightarrow$
  $\rightarrow \forall P_2 \in g :$
  $\mathbf{update}(P_2, V_1, t_1') \cap \mathbf{update}(P_2, V_2, t_2') t_1' < t_2'$

View-synchronous group multicast

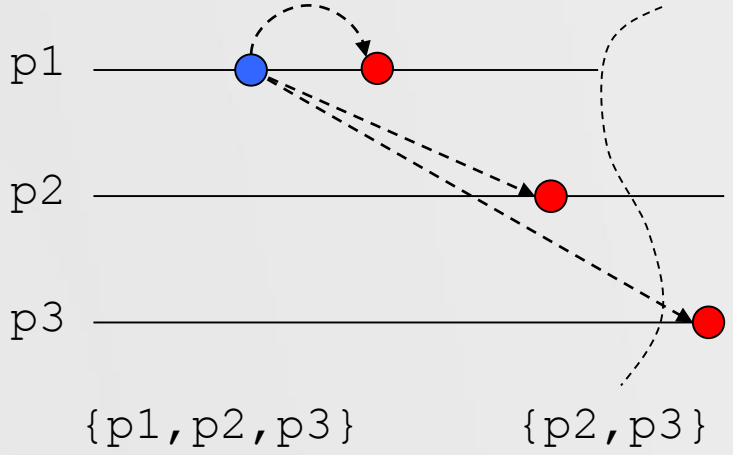
- Συνέπεια παράδοσης μηνυμάτων: αν το μήνυμα m σταλεί στο πλαίσιο της σύνθεσης V, θα παραδοθεί σε όλες τις διεργασίες που εξακολουθούν να είναι μέλη της ομάδας στο πλαίσιο της σύνθεσης V:
- $\mathbf{send}(m)@P_1|V \rightarrow \forall P_2 \in g : \mathbf{recv}(m)@P_2|V$
- Απαιτείται συντονισμός ανάμεσα στην διαχείριση σύνθεσης και την παράδοση μηνυμάτων εφαρμογής.
- Πριν γίνει μετάβαση στην επόμενη σύνθεση, πρέπει να μεταδοθούν/παραδοθούν όλα τα μηνύματα που «ανήκουν» στο πλαίσιο της προηγούμενης σύνθεσης.
- Η συνέπεια παράδοσης αφορά τόσο στην μείωση όσο και στην αύξηση της σύνθεσης της ομάδας.
  - μπορεί να γίνει διαφοροποίηση.

View-synchronous group multicast

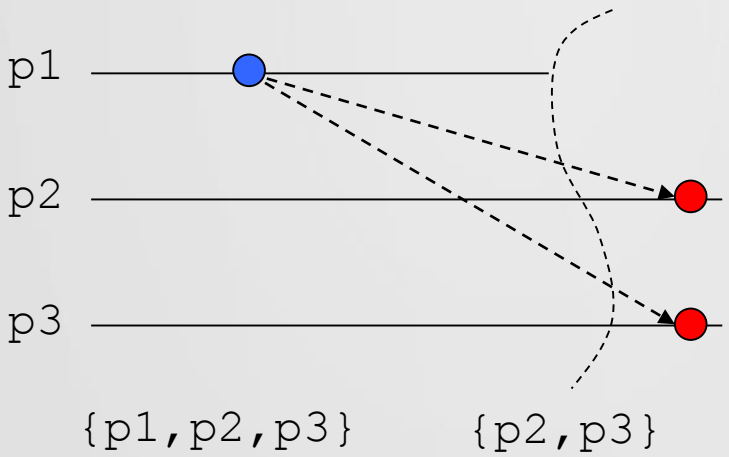
(1) view synchronous



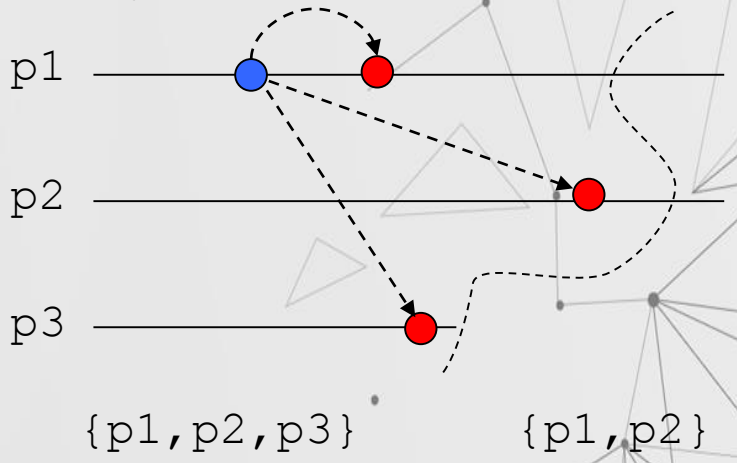
(2) NOT view synchronous



(3) NOT view synchronous



(4) view synchronous

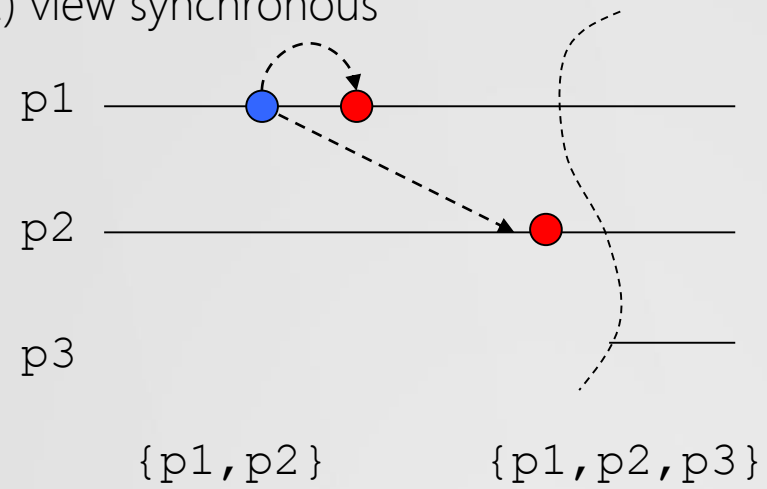


send

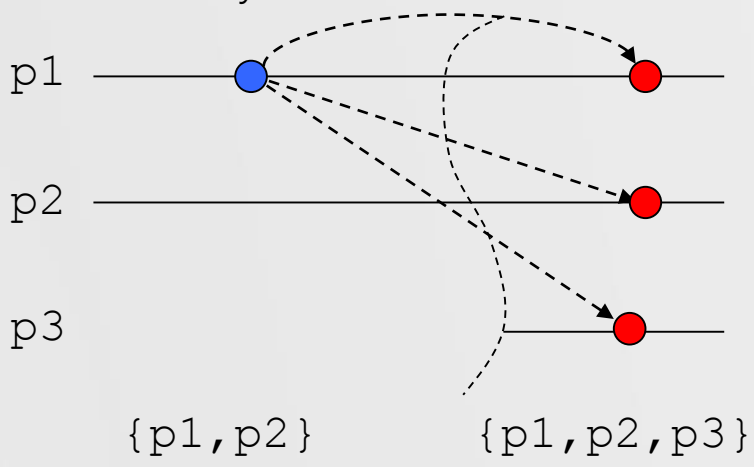
deliver

View-synchronous group multicast

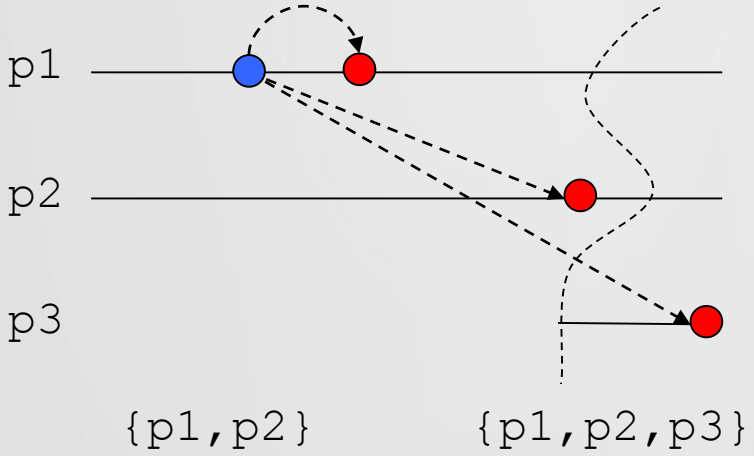
(1) view synchronous



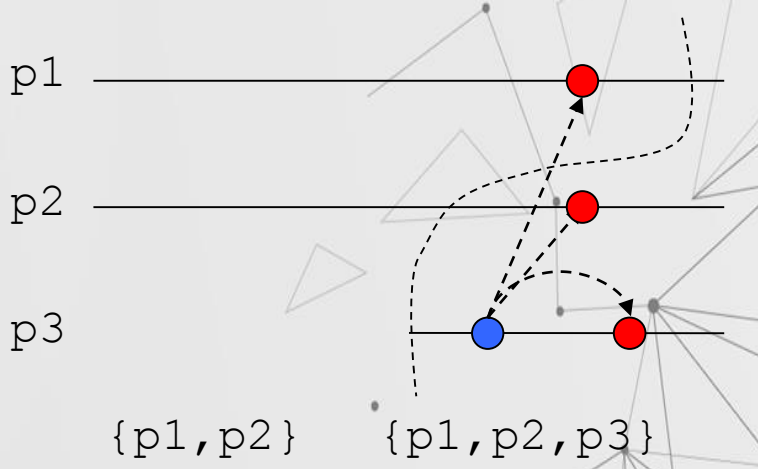
(2) NOT view synchronous



(3) NOT view synchronous



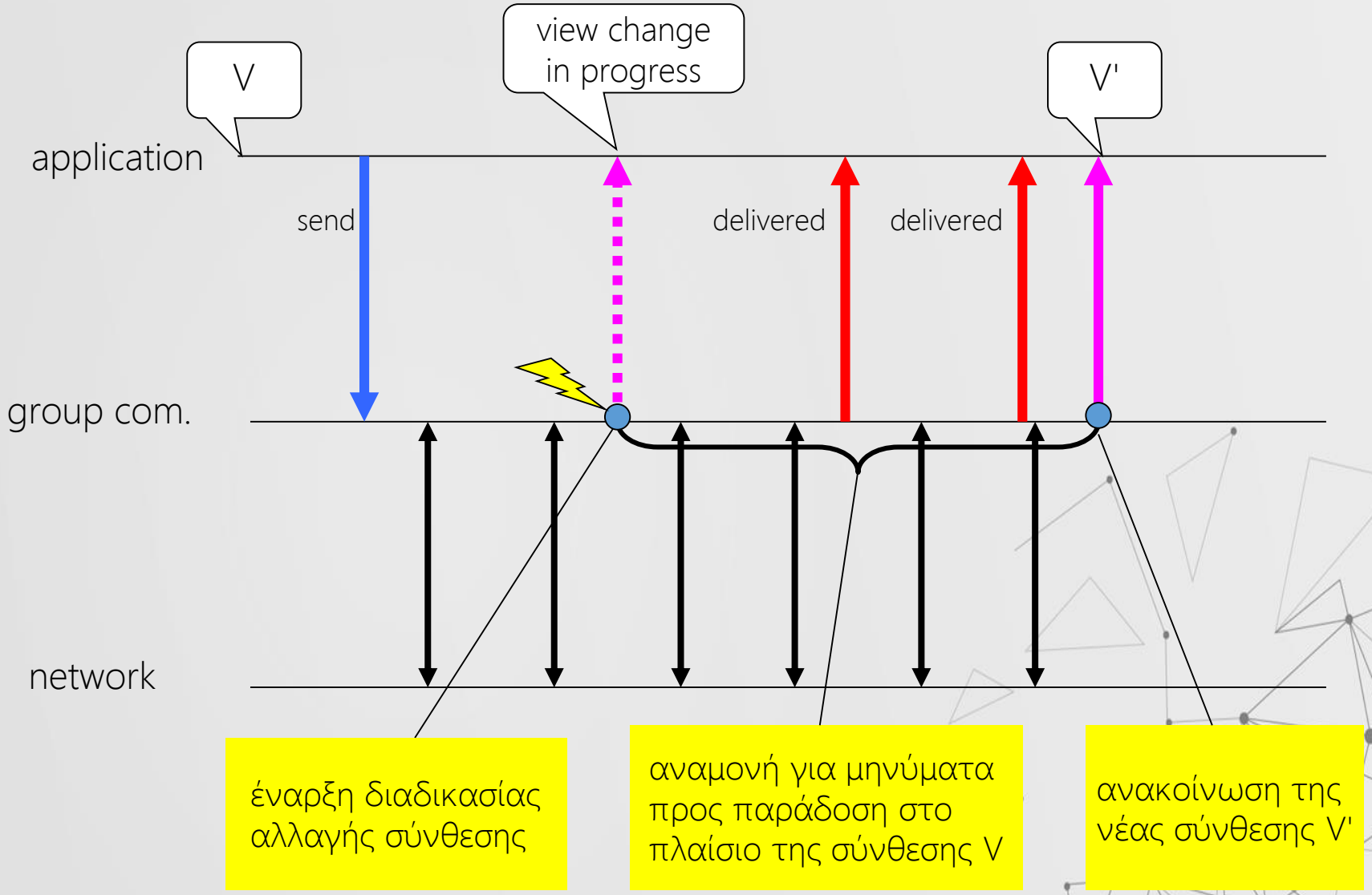
(4) NOT view synchronous



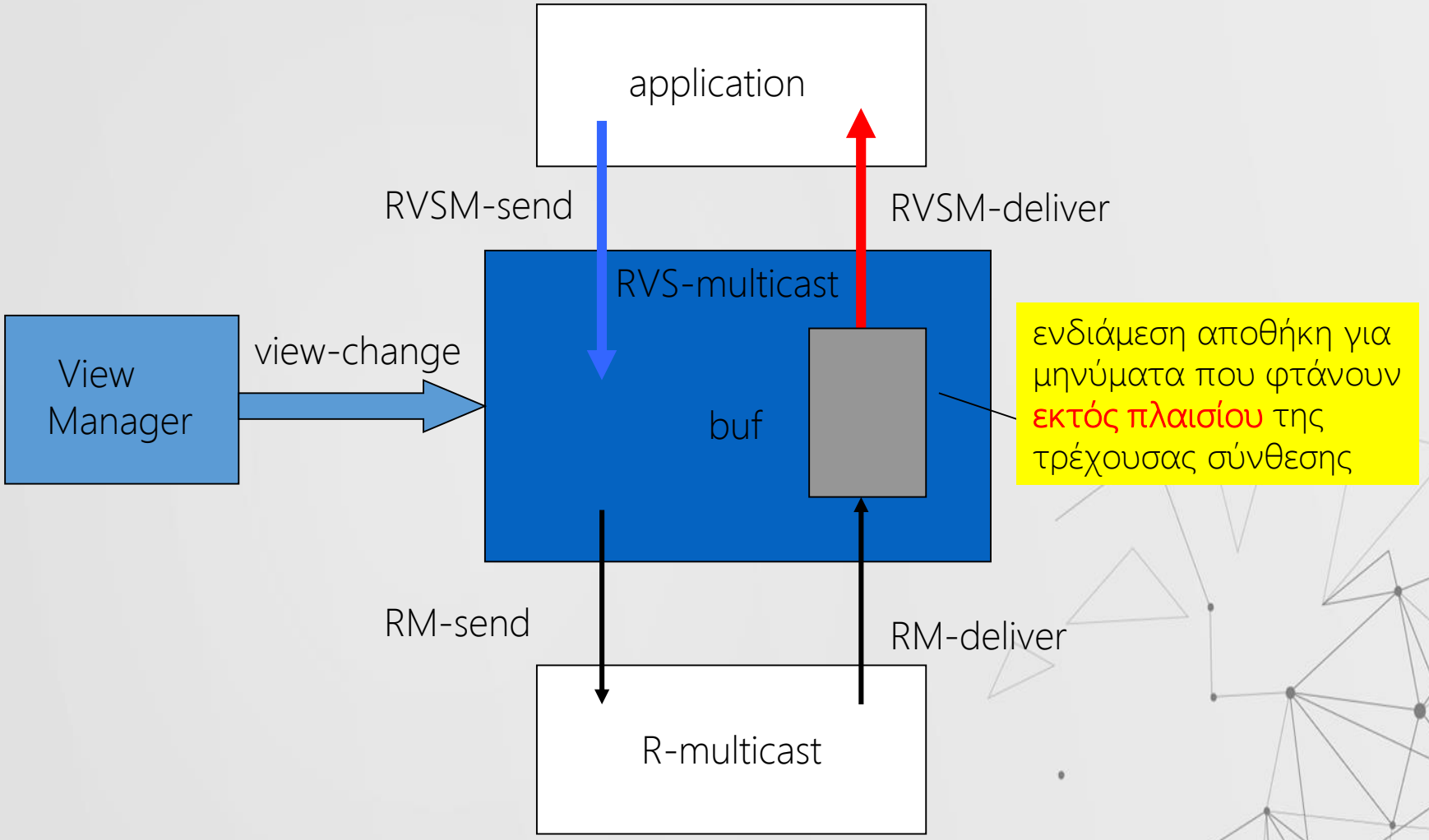
send

deliver

View-synchronous group multicast



View-synchronous group multicast



Υλοποίηση διαχείρισης σύνθεσης

- Κεντρικοποιημένη: μια διακεκριμένη διεργασία «εγκρίνει» τις αλλαγές, και ενημερώνει όλα τα μέλη.
- Συμμετρική: η «έγκριση» των αλλαγών σύνθεσης συναποφασίζεται από όλα τα μέλη της ομάδας, μέσα από κατάλληλο συγχρονισμό (πρωτόκολλο).
- Συνδυασμός: Μπορεί να γίνει συνδυασμός του πρωτοκόλλου απόφασης αλλαγής σύνθεσης και του πρωτοκόλλου μετάδοσης/παράδοσης μηνυμάτων στην ομάδα.

Επιπλέον θέματα υλοποίησης

- Ποια οντότητα/διεργασία «εκπροσωπεί» την ομάδα;
- Τι γίνεται αν η ομάδα είναι «κενή»;
- Πως ανακαλύπτει μια νεοεισερχόμενη διεργασία την ομάδα, και σε ποια διεργασία στέλνει την αίτηση εισαγωγής στην ομάδα;
  - βλέπε γνωστές λύσεις: υπηρεσία καταλόγου, πολυεκπομπή.
- Κατάλληλος χειρισμός βλαβών κατά τη διάρκεια της διαδικασίας αλλαγής/ενημέρωσης σύνθεσης
  - η βλάβη ισοδυναμεί με αυτόματη αποχώρηση από την ομάδα
  - πότε γίνεται η σχετική ανακοίνωση της αλλαγής σύνθεσης;

Σειρά παράδοσης μηνυμάτων

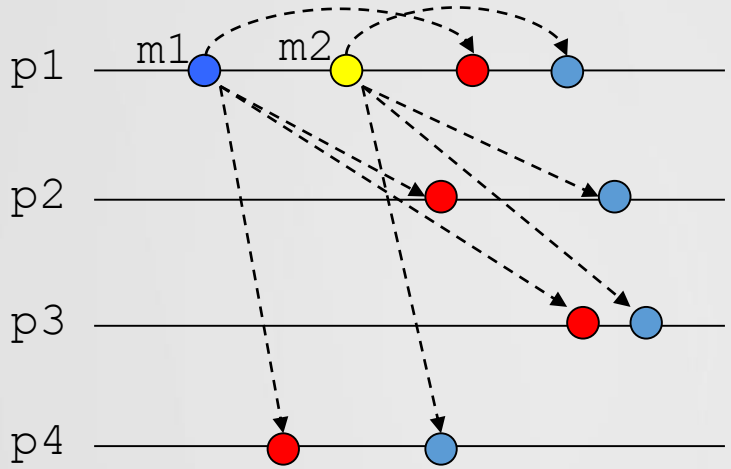
- FIFO order: τα μηνύματα παραδίδονται σε κάθε μέλος με την σειρά με την οποία τα έστειλε ο αποστολέας.
- Αιτιολογική σειρά (causal order): τα μηνύματα παραδίδονται σε κάθε μέλος της ομάδας σύμφωνα με την αιτιολογική σειρά τους.
- Καθολική σειρά (total order): όλα τα μηνύματα παραδίδονται σε κάθε μέλος με την ίδια σειρά

Βασική αρχή λειτουργίας

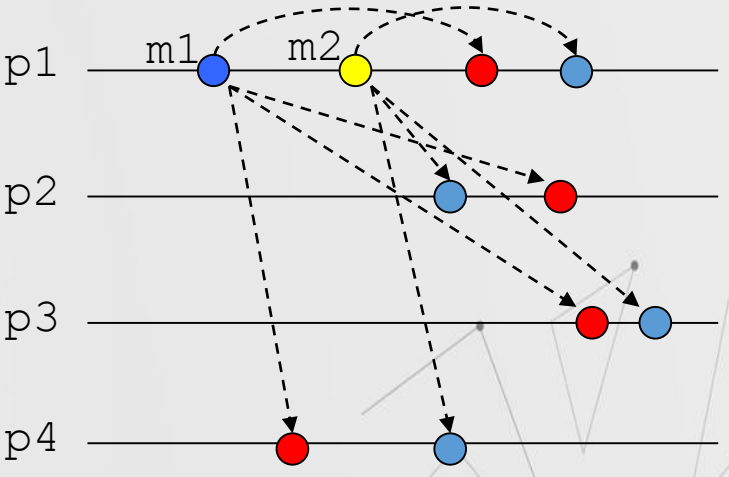
- Κάθε μήνυμα που φτάνει από το δίκτυο (το όποιο στρώμα επικοινωνίας/μεταφοράς χρησιμοποιείται) τοποθετείται σε μια ενδιάμεση αποθήκη.
- Ένα μήνυμα βγαίνει από την αποθήκη και παραδίδεται στην εφαρμογή όταν είναι σίγουρο πως έχουν ήδη παραδοθεί όλα τα «προηγούμενα» μηνύματα.
- Λέμε ότι το μήνυμα είναι πλέον σταθερό (stable).
- Οι απαιτήσεις για την σειρά παράδοσης εξαρτώνται από το είδος της επιθυμητής σειριοποίησης.

FIFO Order

FIFO-Multicast



NOT FIFO-Multicast

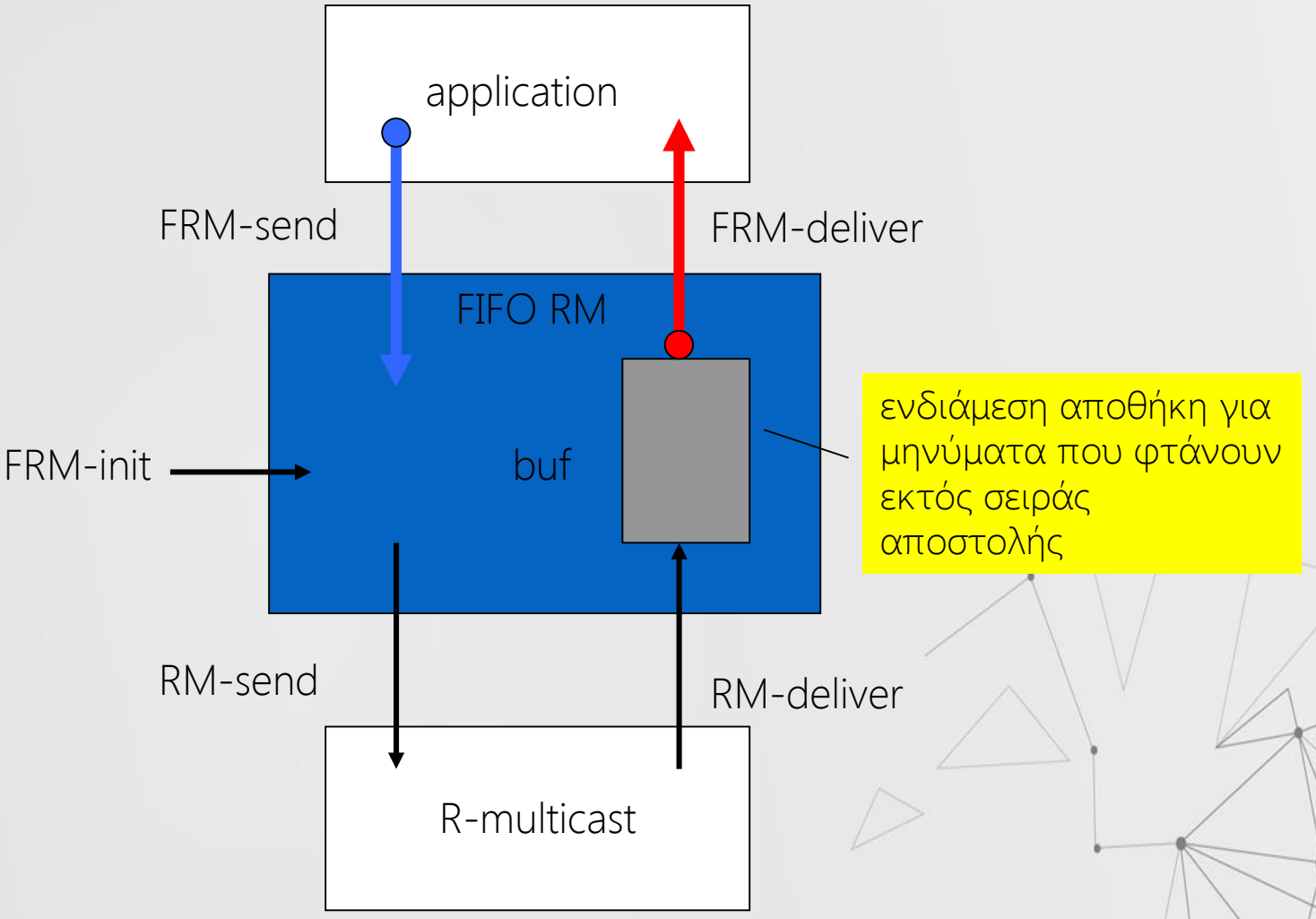


send(m1)
send(m2)
deliver(m1)
deliver(m2)

FIFO reliable multicast (FRM)

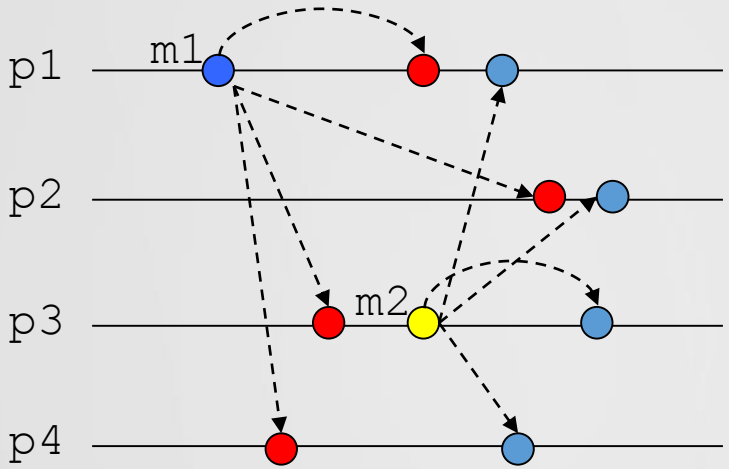
- Σε έναν μετρητή $d[pid]$ αποθηκεύεται ο σειριακός αριθμός του τελευταίου μηνύματος που έστειλε η διεργασία pid και έχει ήδη παραδοθεί στην εφαρμογή.
- Τα μηνύματα που καταφθάνουν από το δίκτυο τοποθετούνται σε ενδιάμεση αποθήκη, μέχρι να μπορεί να παραδοθούν με την σωστή σειρά.
- Η αποθήκη ελέγχεται για το αν υπάρχει μήνυμα m με αποστολέα pid και σειριακό αριθμό $k == d[pid] + 1$.
- Τότε το m απομακρύνεται από την αποθήκη, παραδίδεται στην εφαρμογή, και $d[pid] = d[pid] + 1$.

FIFO reliable multicast (FRM)

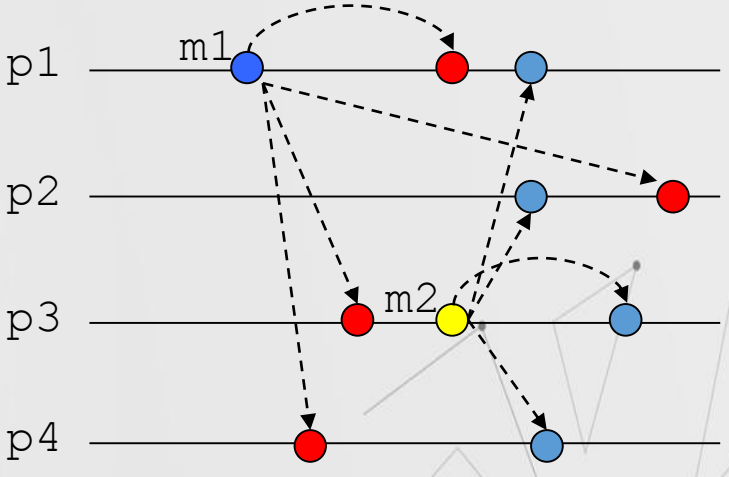


Causal Order

Causal-Multicast



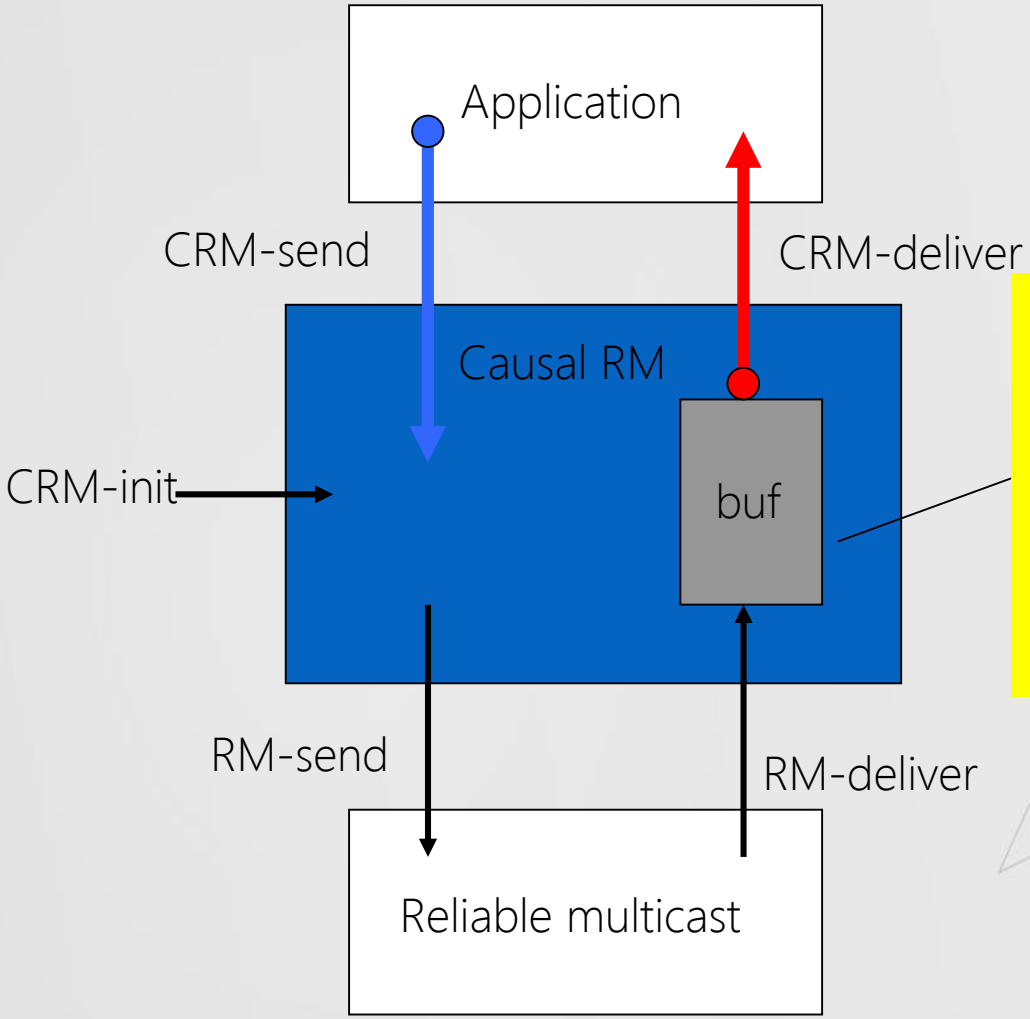
NOT Causal-Multicast



Causal reliable multicast (CRM, 1)

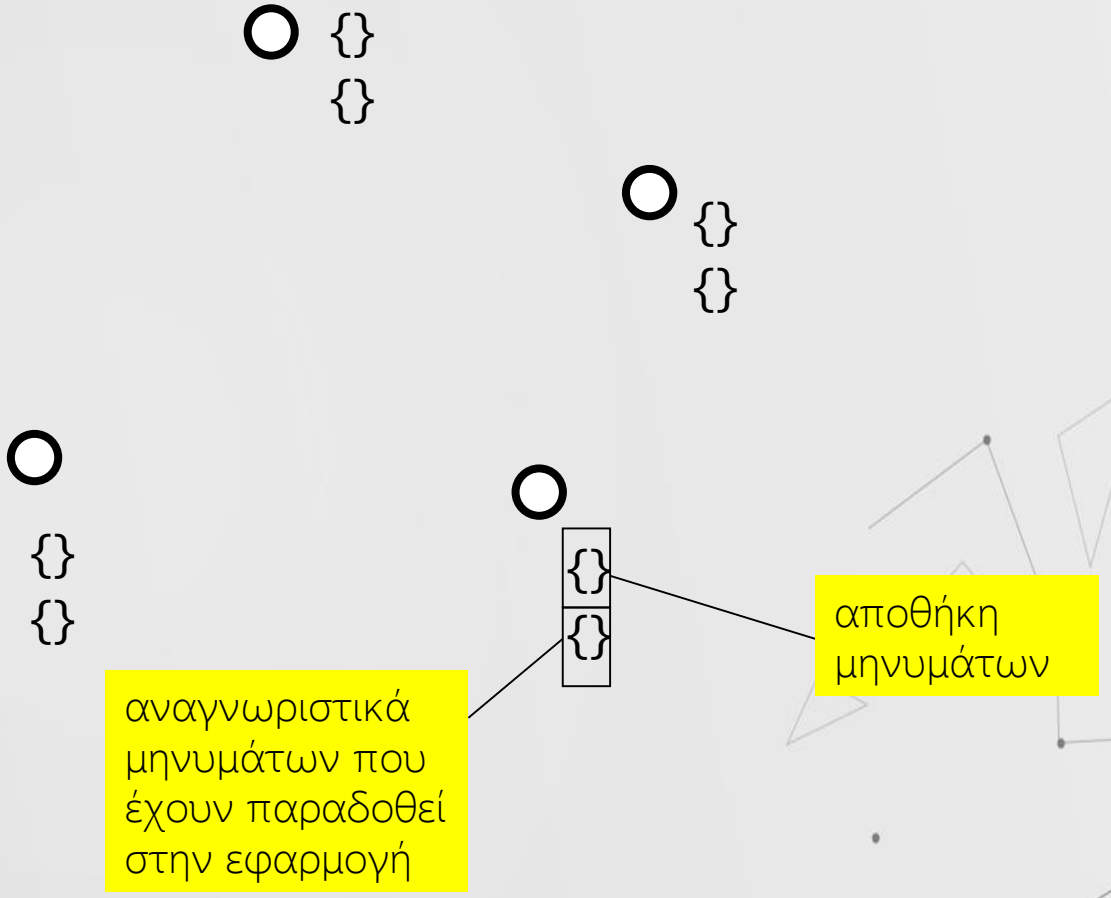
- Τα μηνύματα που καταφθάνουν από το δίκτυο τοποθετούνται σε ενδιάμεση αποθήκη, μέχρι να μπορεί να παραδοθούν με την σωστή σειρά.
- Πως καταγράφεται η λογική σειρά των μηνυμάτων;
- Επισυνάπτεται ο σειριακός αριθμός (για FIFO) και … τα αναγνωριστικά όσων μηνυμάτων είχαν παραδοθεί στην εφαρμογή όταν δημιουργήθηκε το μήνυμα (ως «λογική προϊστορία» του μηνύματος).

Causal reliable multicast (CRM, 2)

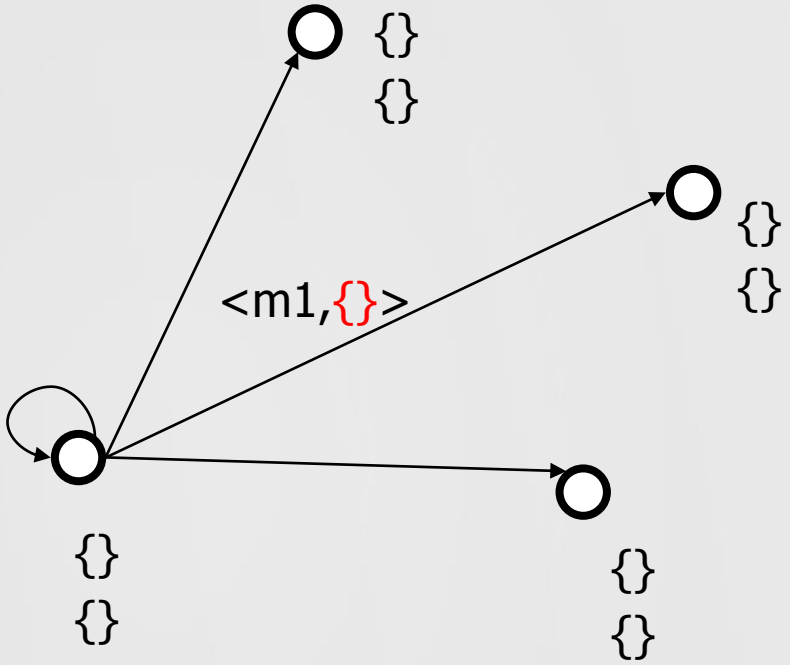


κάθε μήνυμα κρατείται σε ενδιάμεση αποθήκη μέχρι να είναι σίγουρο πως έχουν παραδοθεί στην εφαρμογή όλα τα (λογικά) προηγούμενα μηνύματα που έχουν σταλεί στην ομάδα

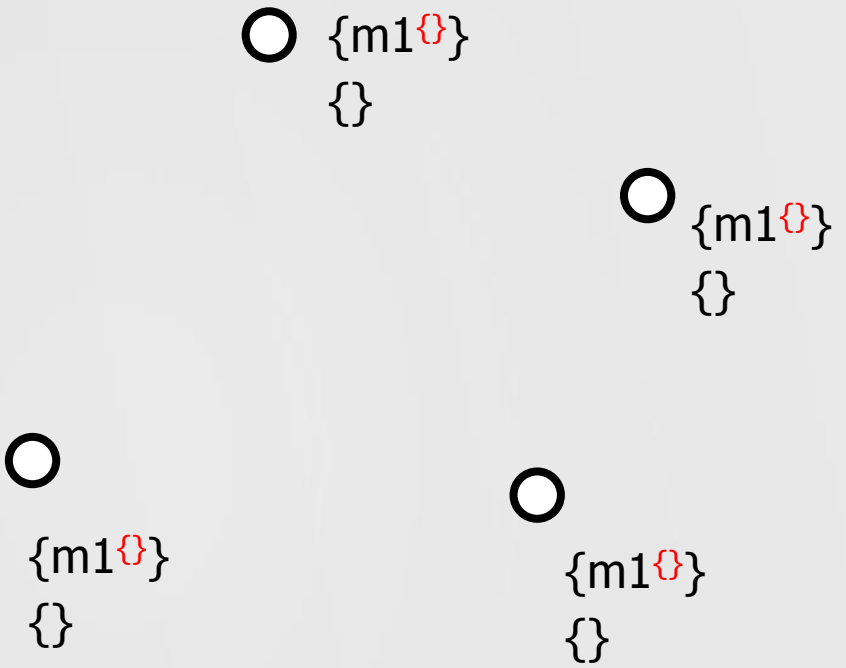
Causal reliable multicast (CRM, 3)



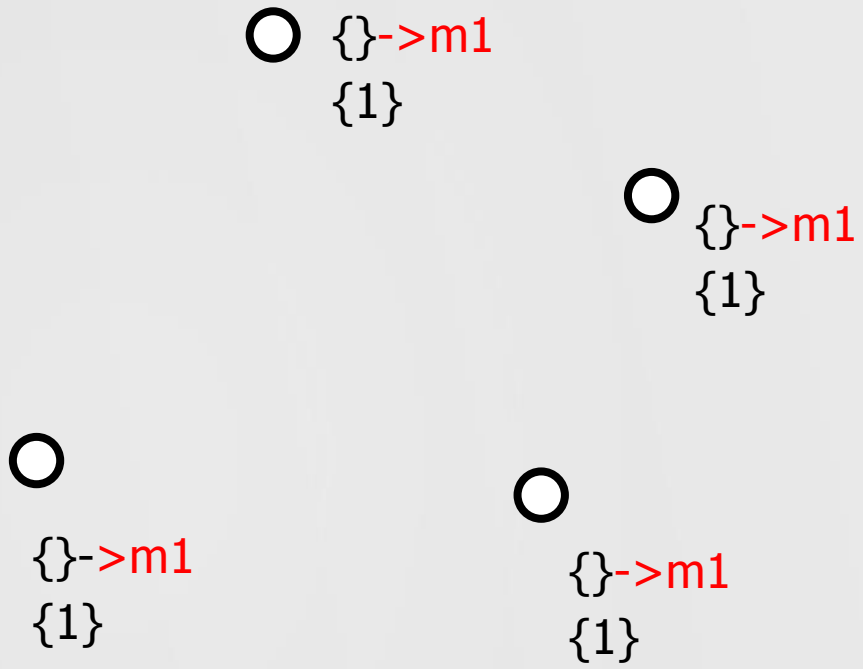
Causal reliable multicast (CRM, 4)



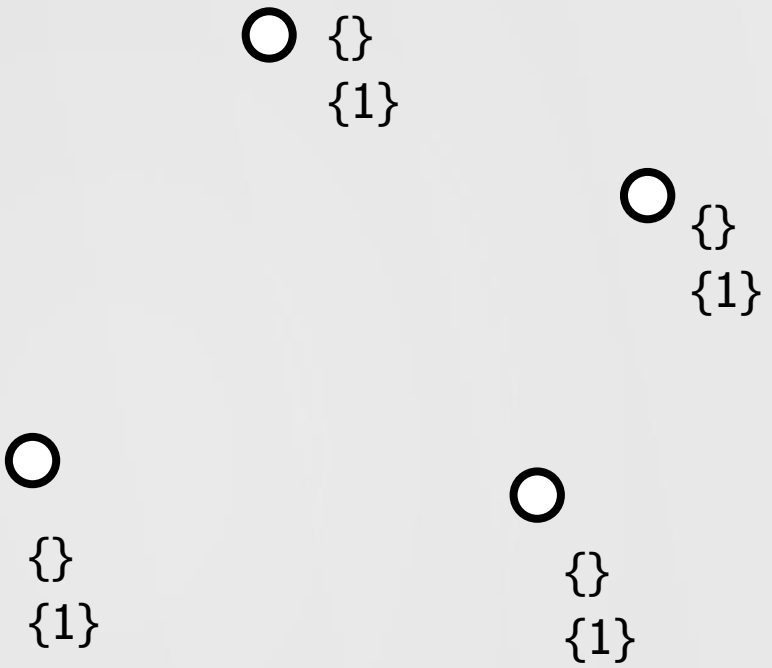
Causal reliable multicast (CRM, 5)



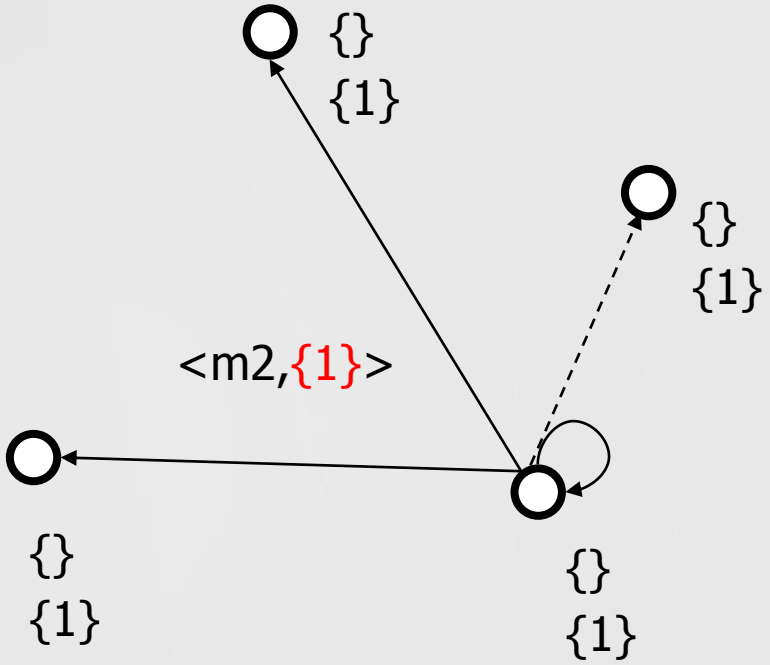
Causal reliable multicast (CRM, 6)



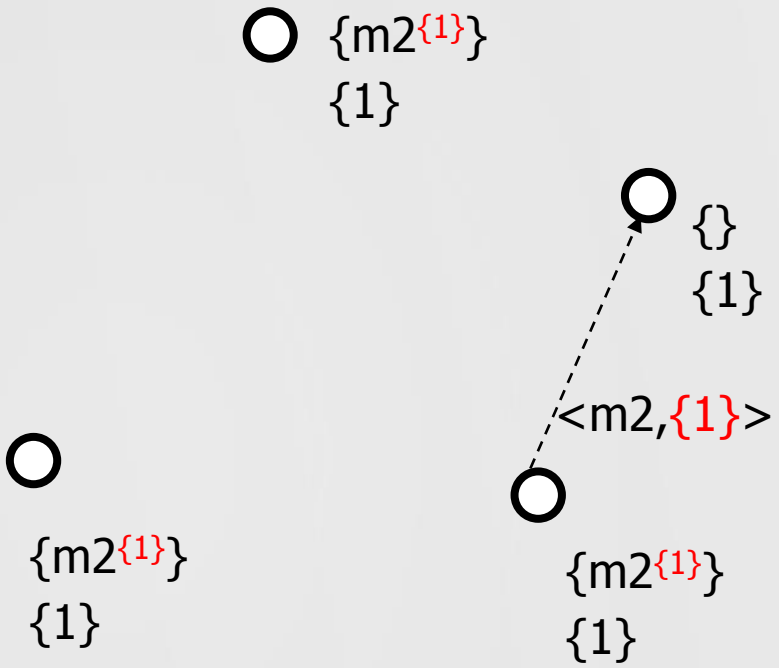
Causal reliable multicast (CRM, 7)



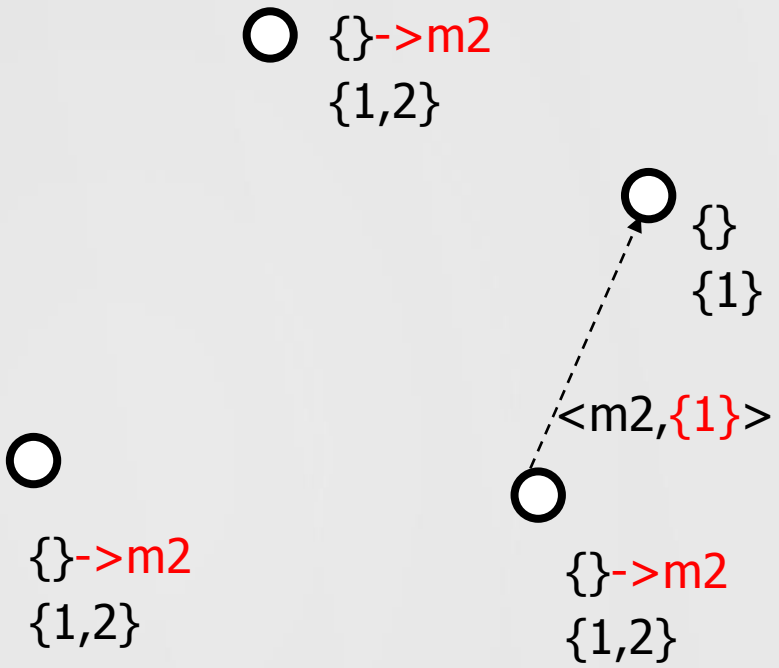
Causal reliable multicast (CRM, 8)



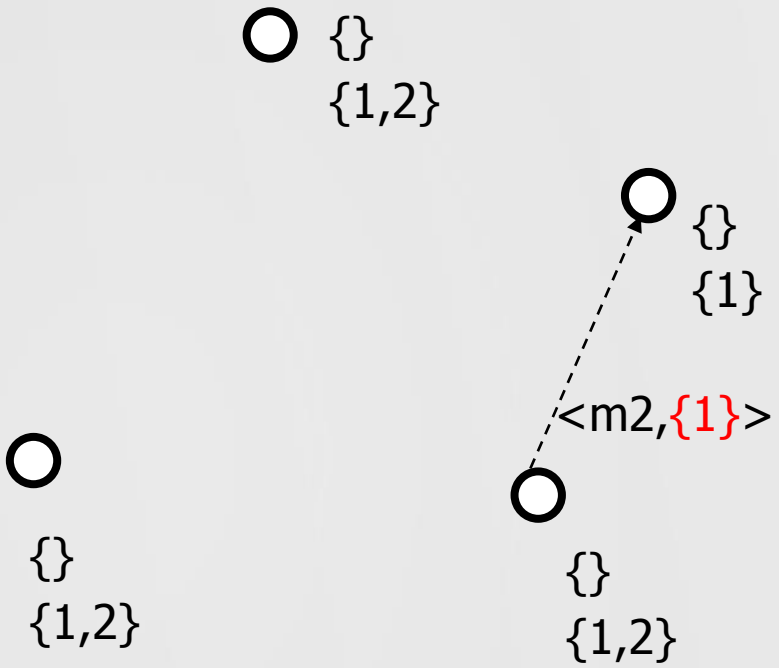
Causal reliable multicast (CRM, 9)



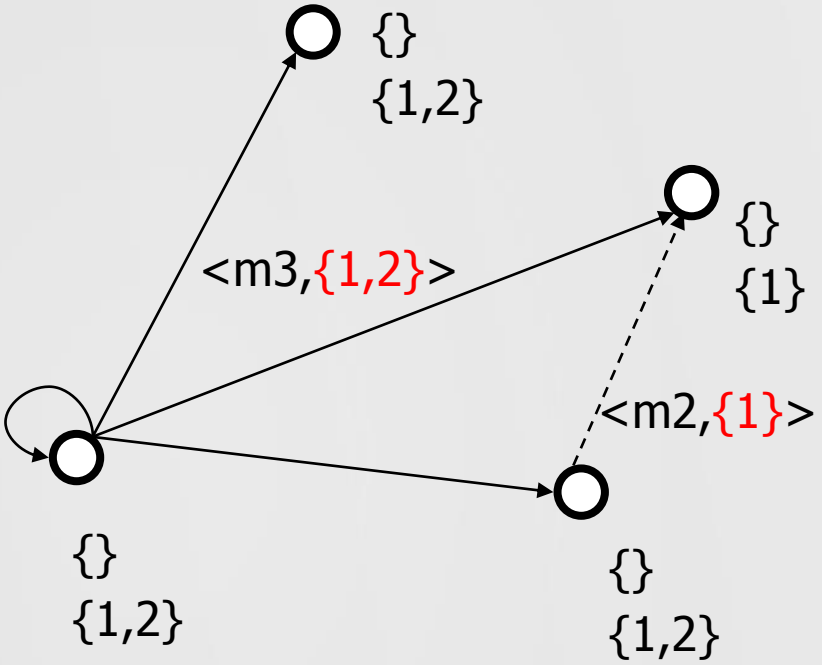
Causal reliable multicast (CRM, 10)



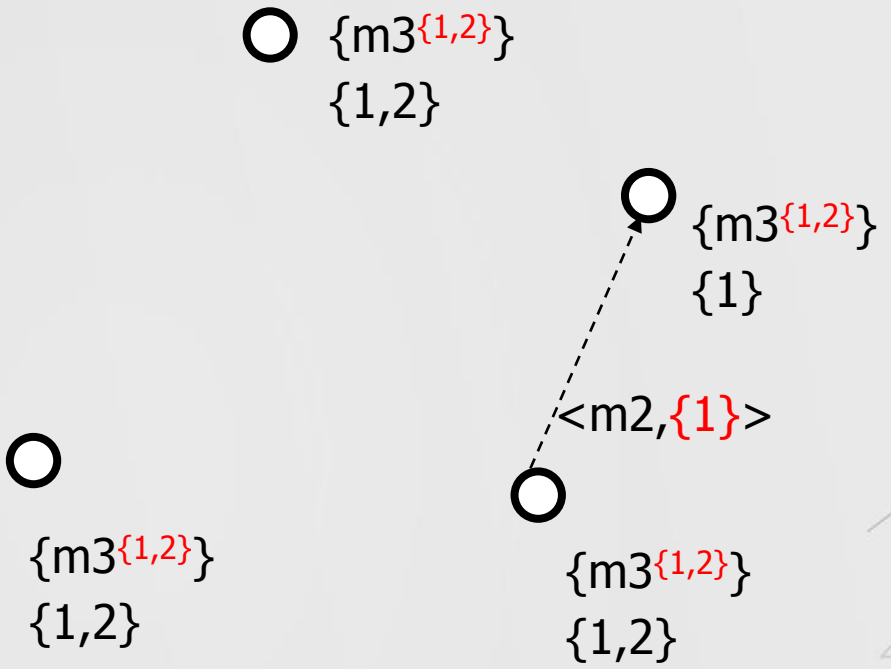
Causal reliable multicast (CRM, 11)



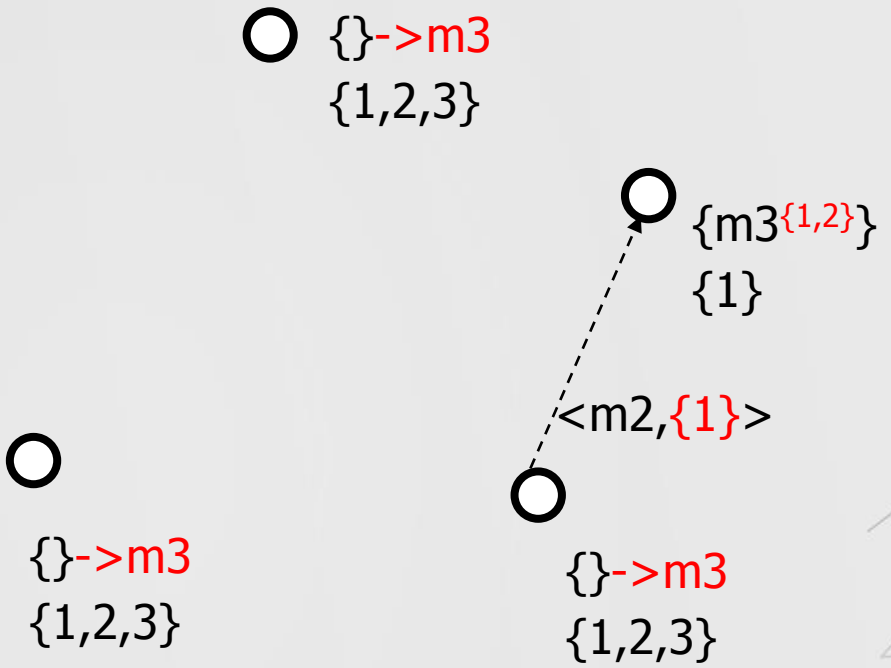
Causal reliable multicast (CRM, 12)



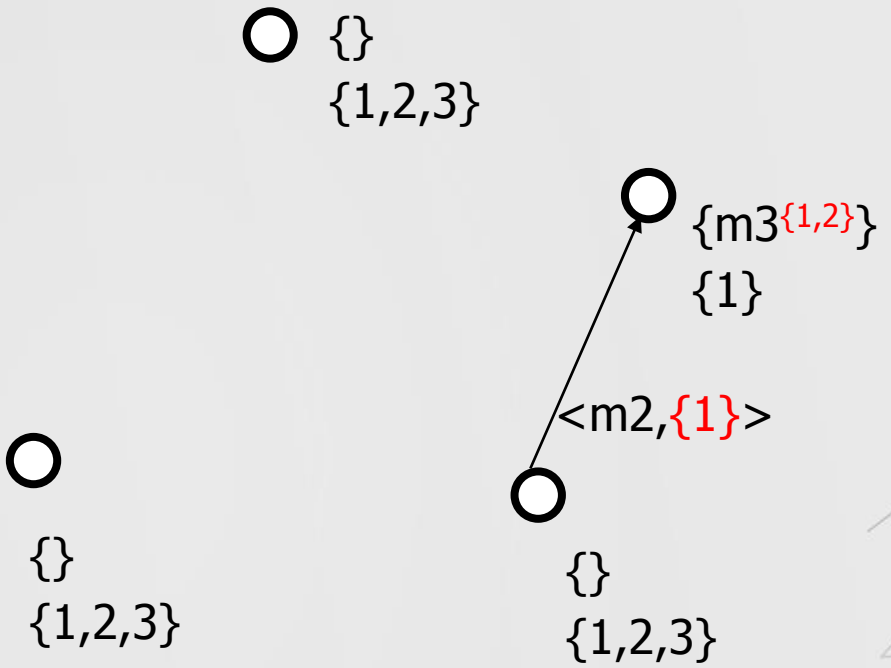
Causal reliable multicast (CRM, 13)



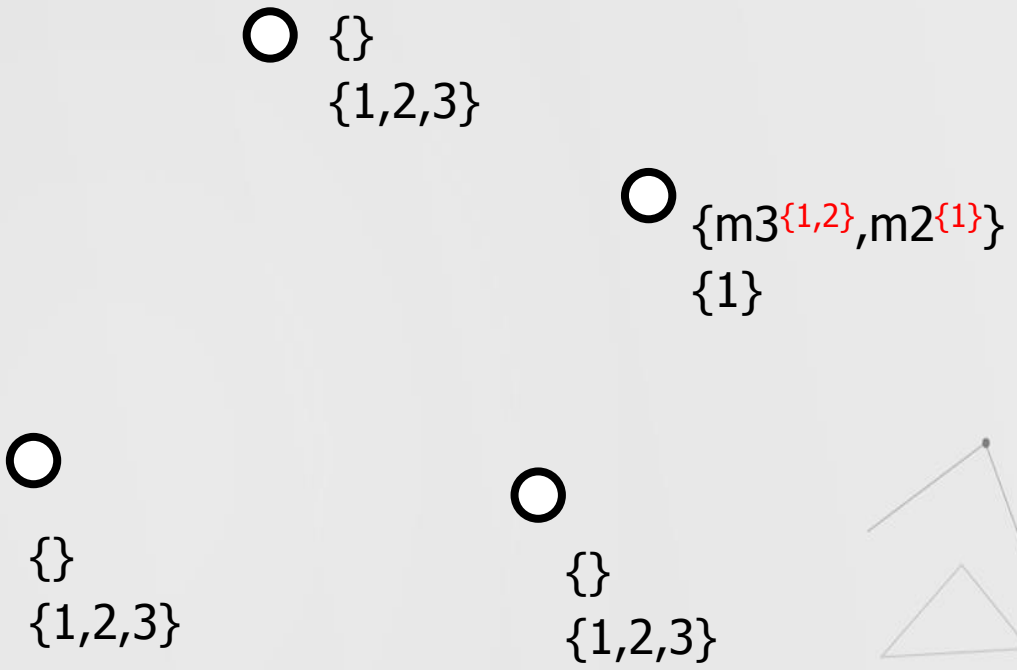
Causal reliable multicast (CRM, 14)



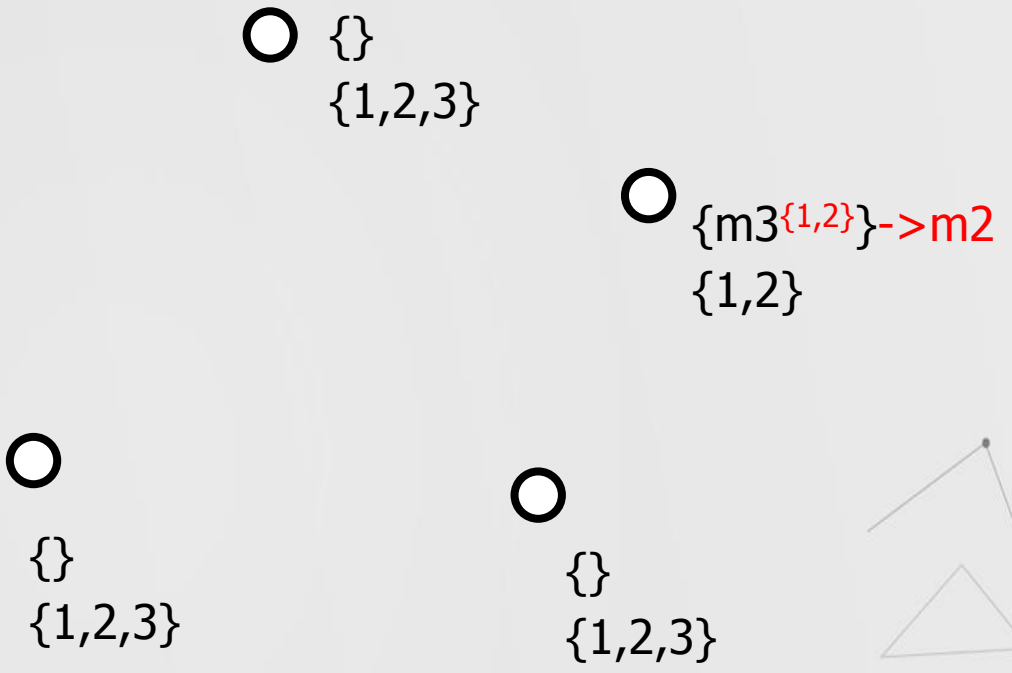
Causal reliable multicast (CRM, 15)



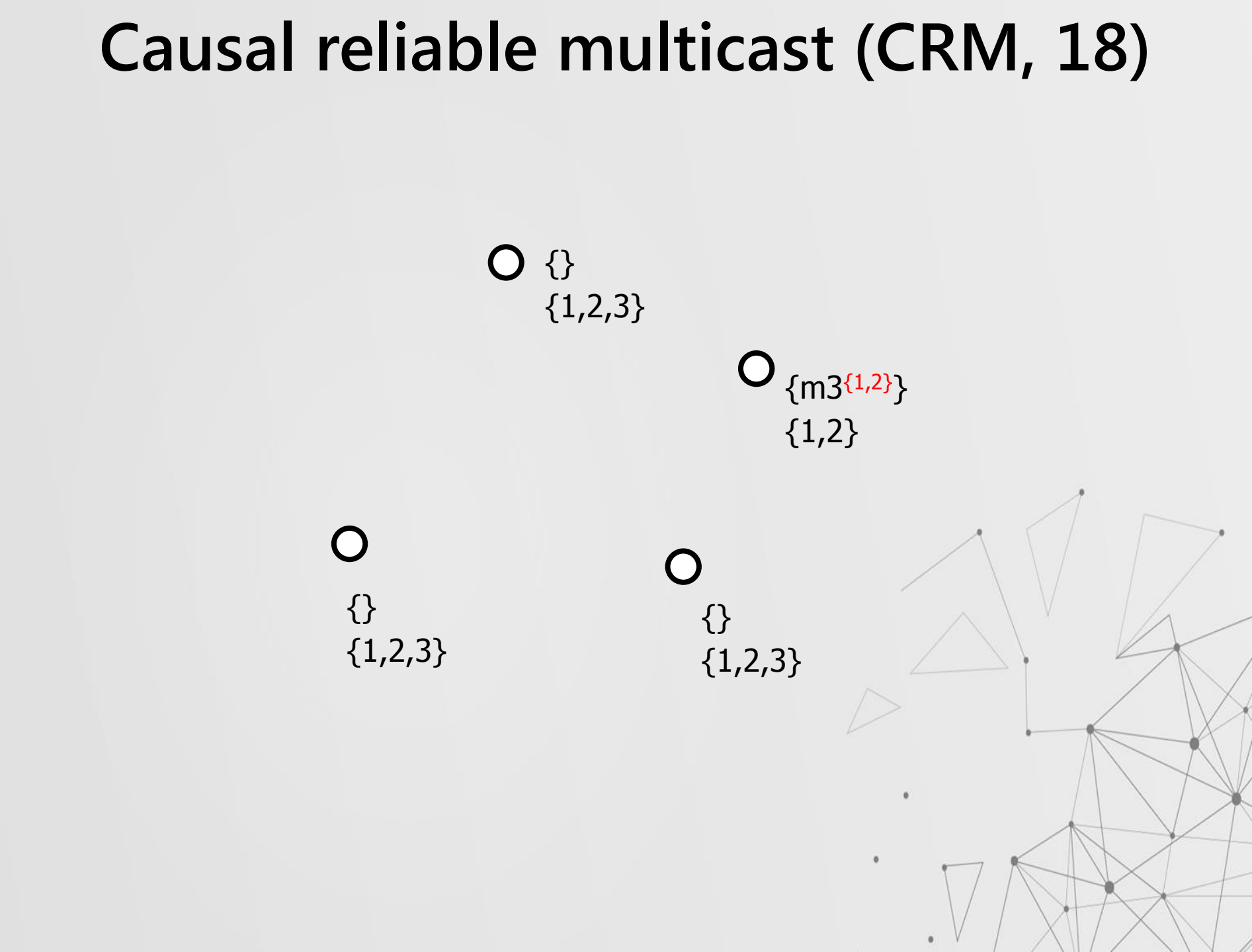
Causal reliable multicast (CRM, 16)



Causal reliable multicast (CRM, 17)

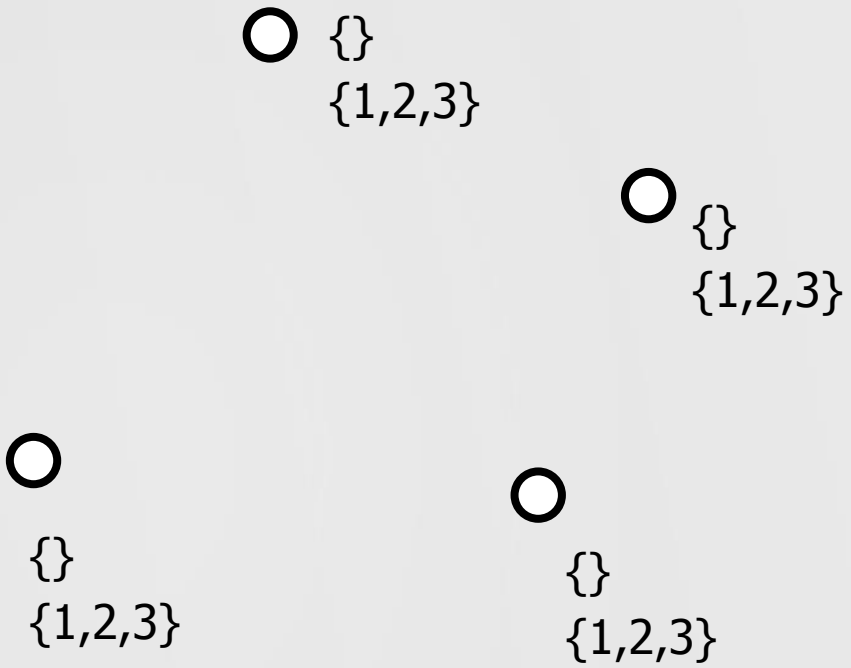


Causal reliable multicast (CRM, 18)



Causal reliable multicast (CRM, 19)

Causal reliable multicast (CRM, 20)

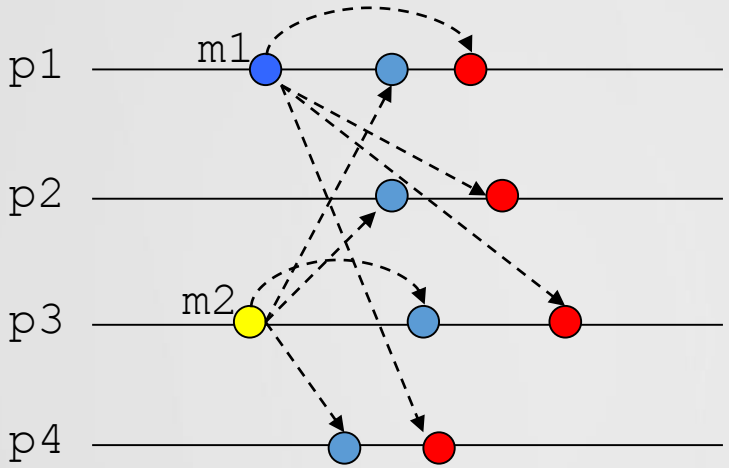


Βελτιστοποίηση

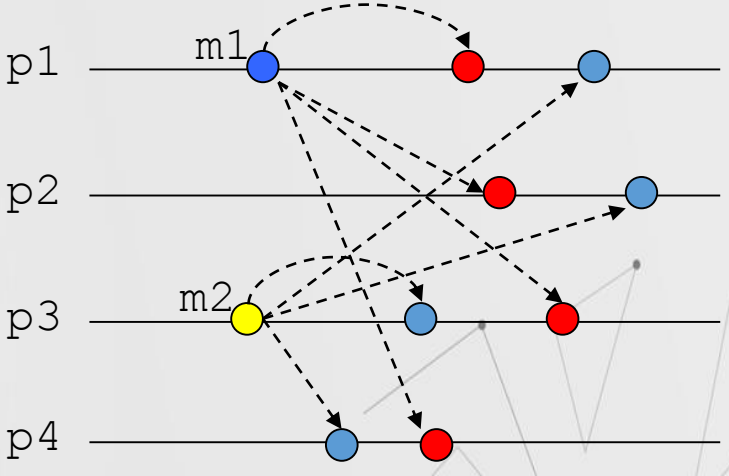
- Ο αριθμός των μηνυμάτων που παραδίδονται στην εφαρμογή, και άρα ο αριθμός των αναγνωριστικών που επισυνάπτονται σε κάθε μήνυμα μεγαλώνει συνεχώς.
- Η περισσότερη πληροφορία είναι περιττή καθώς η σχέση λογικής εξάρτησης είναι μεταβατική.
- Αν το m1 προηγείται του m2, και το m2 προηγείται του m3, αρκεί στο m3 να επισυναφθεί το αναγνωριστικό του m2 (δεν χρειάζεται και το αναγνωριστικό του m1).
- Επισυνάπτονται μόνο τα αναγνωριστικά των πιο πρόσφατων λογικά προηγούμενων μηνυμάτων.
- Το πολύ Ν αναγνωριστικά, ένα από κάθε διεργασία.

Total order

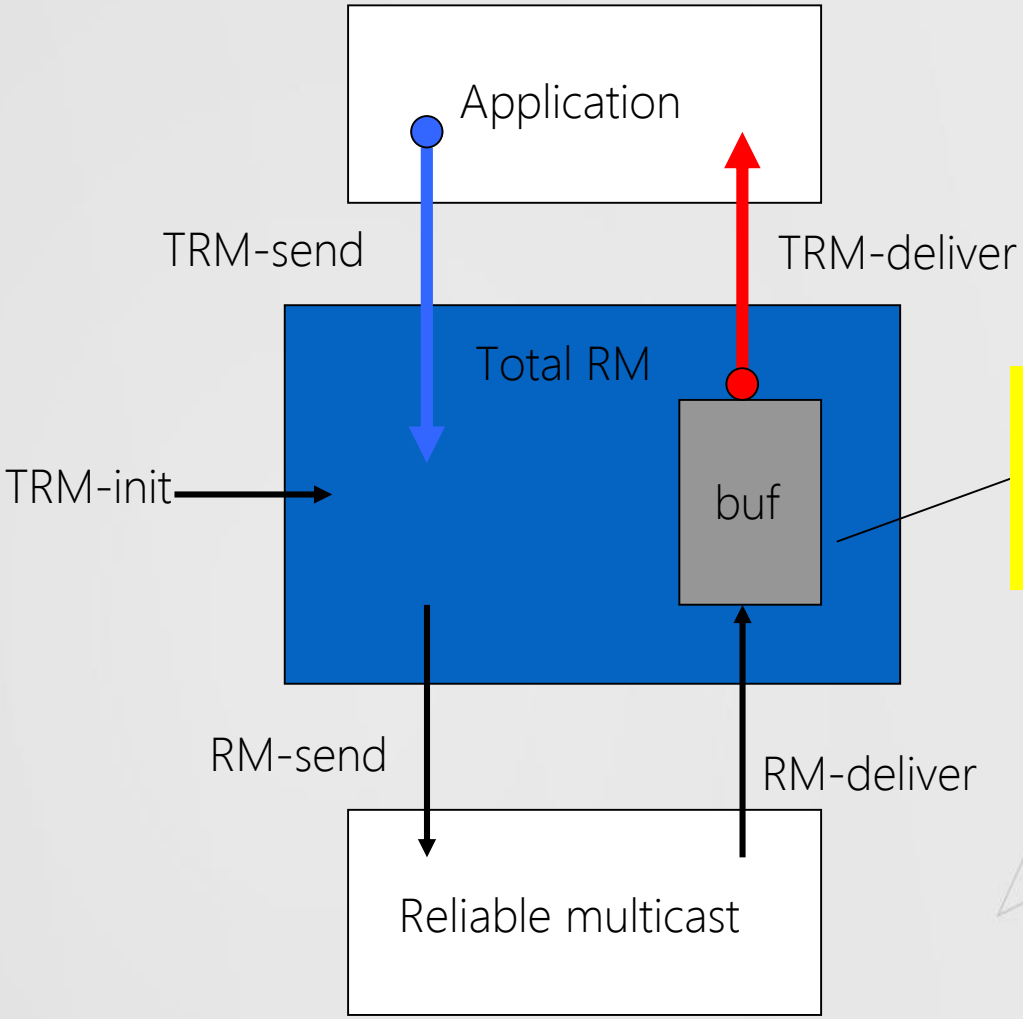
Total-Multicast



NOT Total-Multicast



Total order



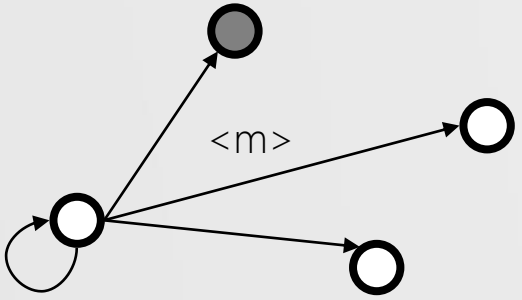
κάθε μήνυμα κρατείται σε ενδιάμεση αποθήκη μέχρι να είναι σίγουρο πως έχει έρθει η σειρά του

Πρωτόκολλο με συντονιστή (1)

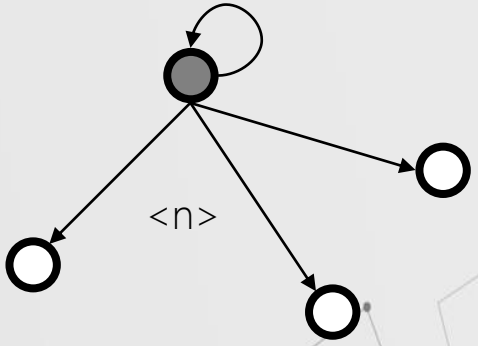
- Κάθε μήνυμα στέλνεται σε όλα τα μέλη της ομάδας.
- Τα μηνύματα που καταφθάνουν από το δίκτυο τοποθετούνται σε ενδιάμεση αποθήκη.
- Μια διακεκριμένη διεργασία - συντονιστής αναθέτει σε κάθε μήνυμα που λαμβάνει έναν σειριακό αριθμό και τον στέλνει/ανακοινώνει στα μέλη, και στην ίδια.
- Τα μηνύματα αφαιρούνται από την αποθήκη και παραδίδονται στην εφαρμογή με βάση τον σειριακό αριθμό που τους αναθέτει ο συντονιστής.
- Αφού οι σειριακοί αριθμοί δίνονται από την ίδια διεργασία, επιτυγχάνεται κοινή σειρά παράδοσης.

Πρωτόκολλο με συντονιστή (2)

1:
ένα νέο μήνυμα στέλνεται σε όλα τα μέλη της ομάδας

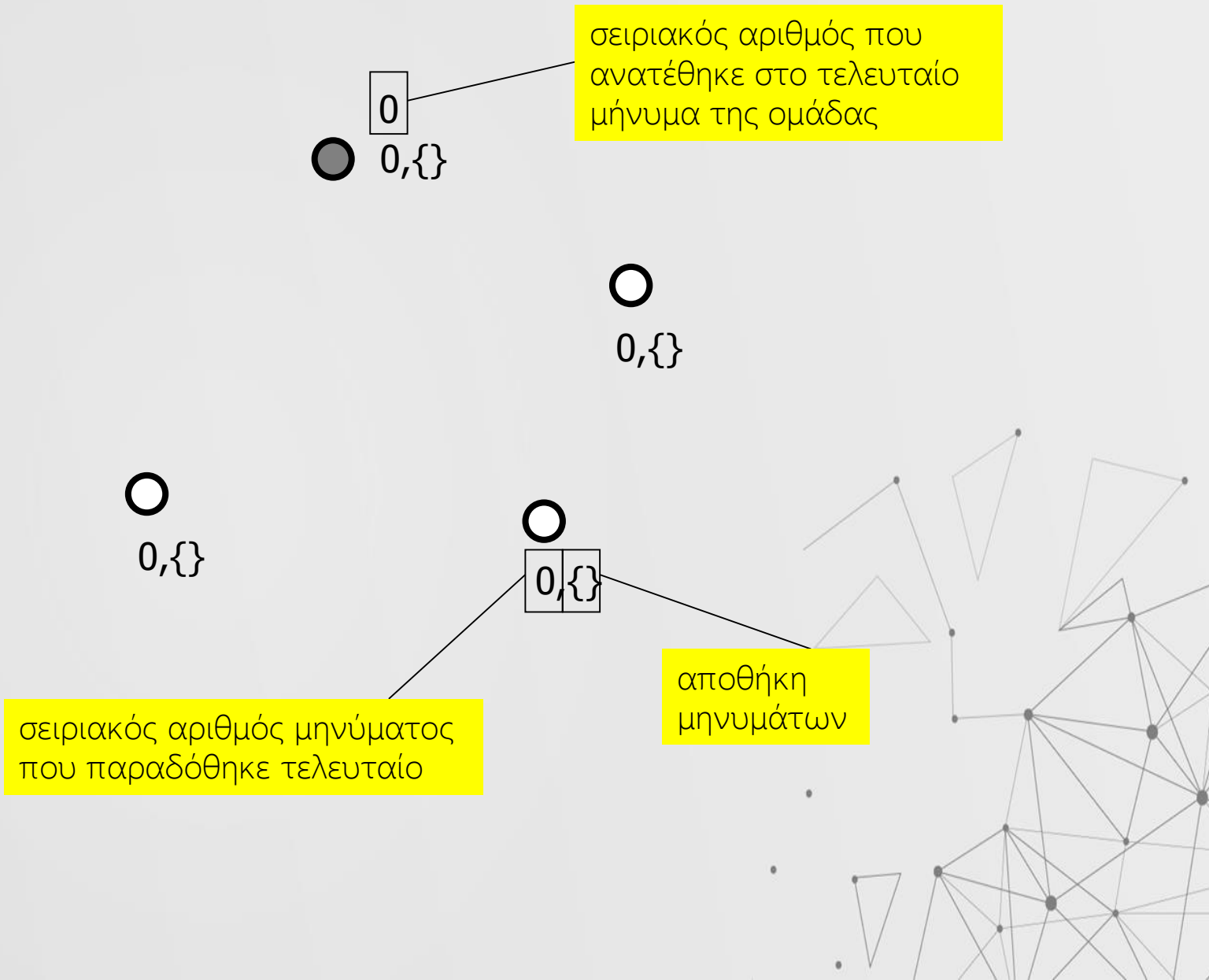


2:
ο συντονιστής στέλνει τον σειριακό αριθμό παράδοσης για αυτό το μήνυμα σε όλα τα μέλη

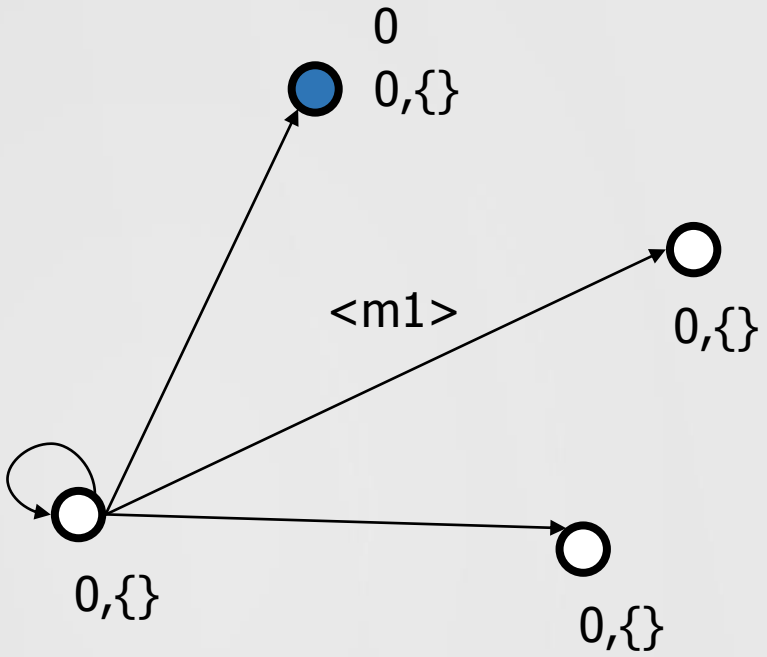


3:
κάθε μέλος παραδίδει τα μηνύματα που λαμβάνει στην τοπική εφαρμογή σύμφωνα με τους σειριακούς αριθμούς που στέλνει ο συντονιστής

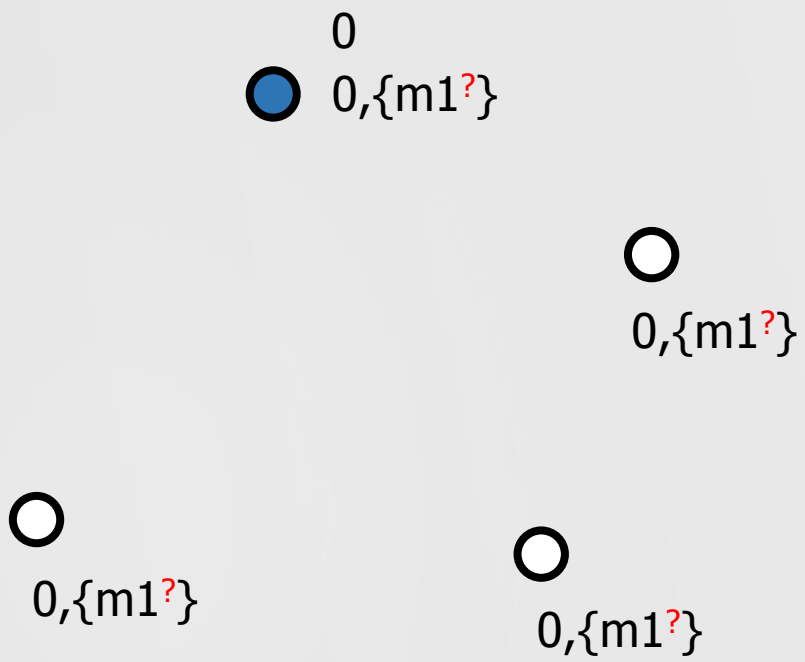
Πρωτόκολλο με συντονιστή (3)



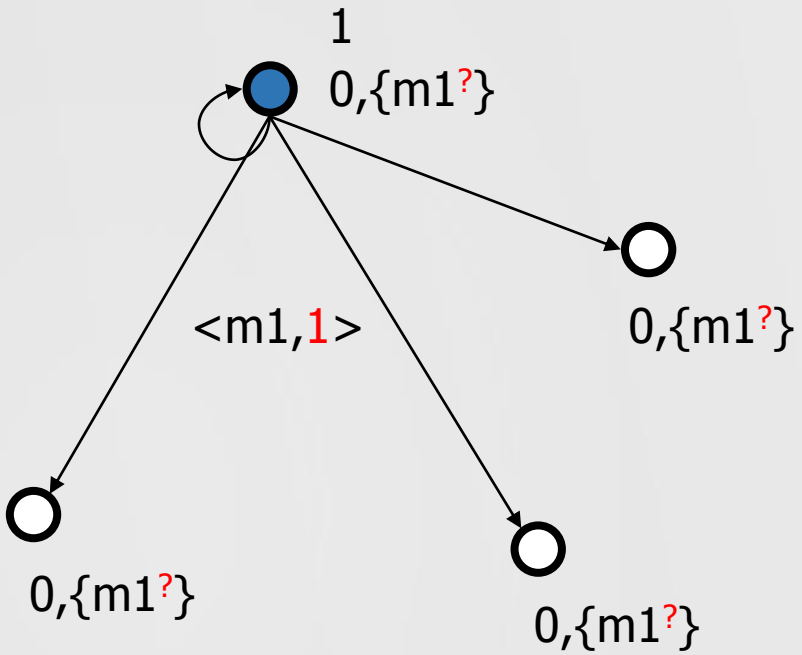
Πρωτόκολλο με συντονιστή (4)



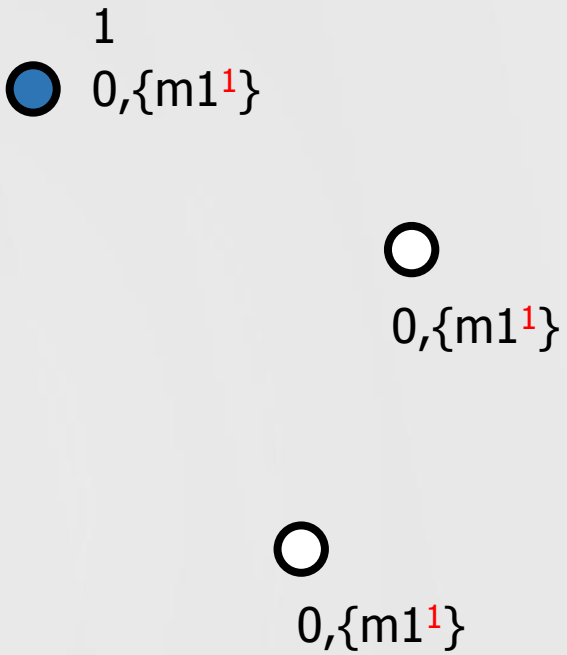
Πρωτόκολλο με συντονιστή (5)



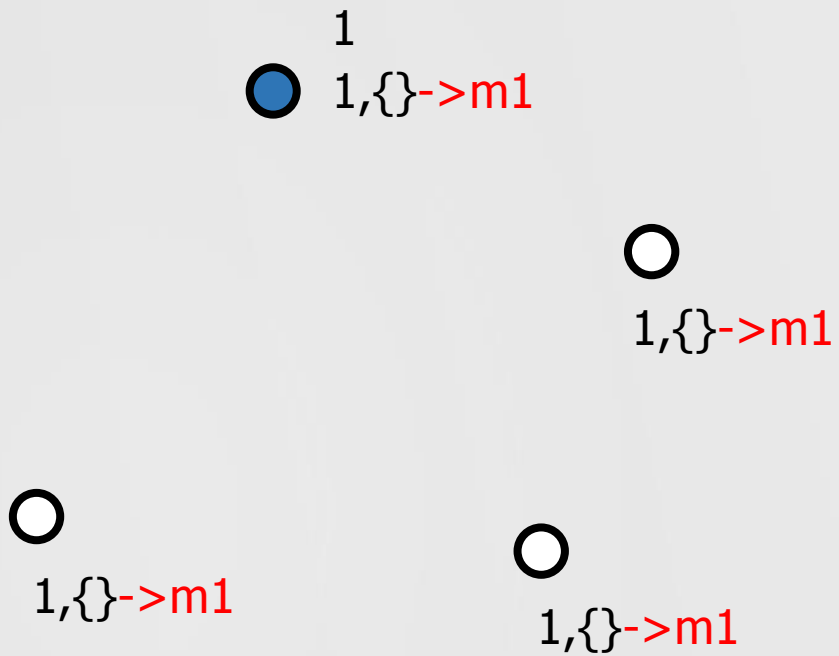
Πρωτόκολλο με συντονιστή (6)



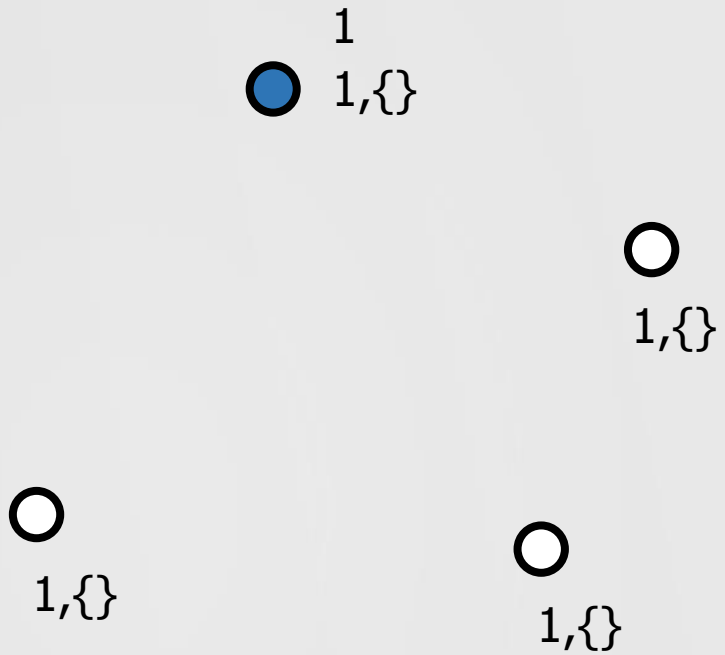
Πρωτόκολλο με συντονιστή (7)



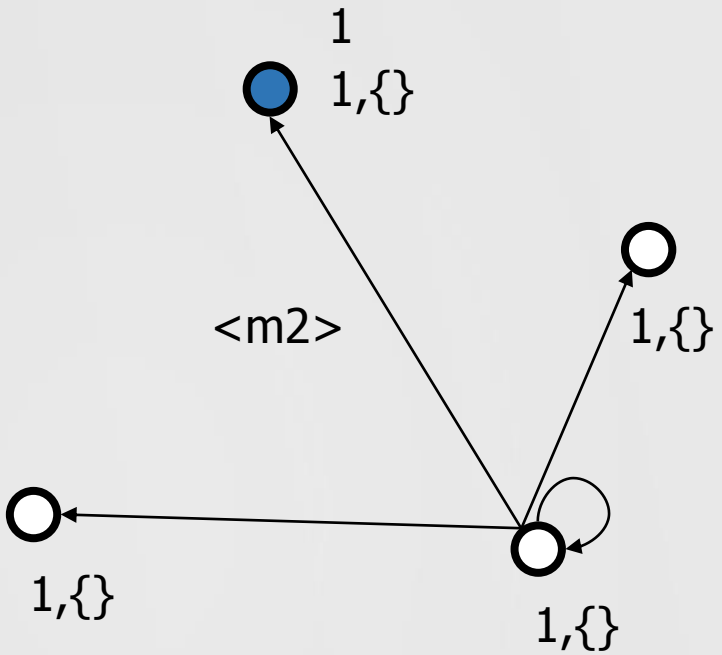
Πρωτόκολλο με συντονιστή (8)



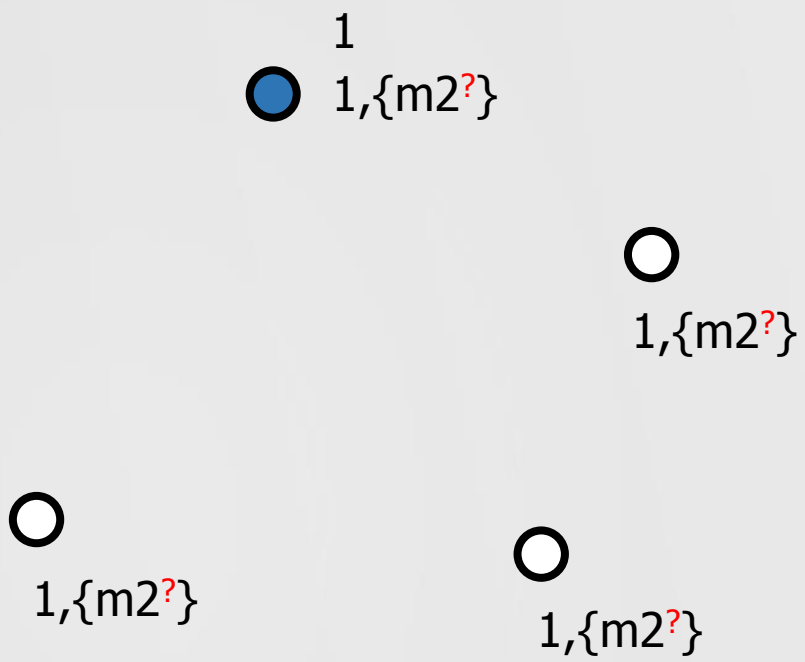
Πρωτόκολλο με συντονιστή (9)



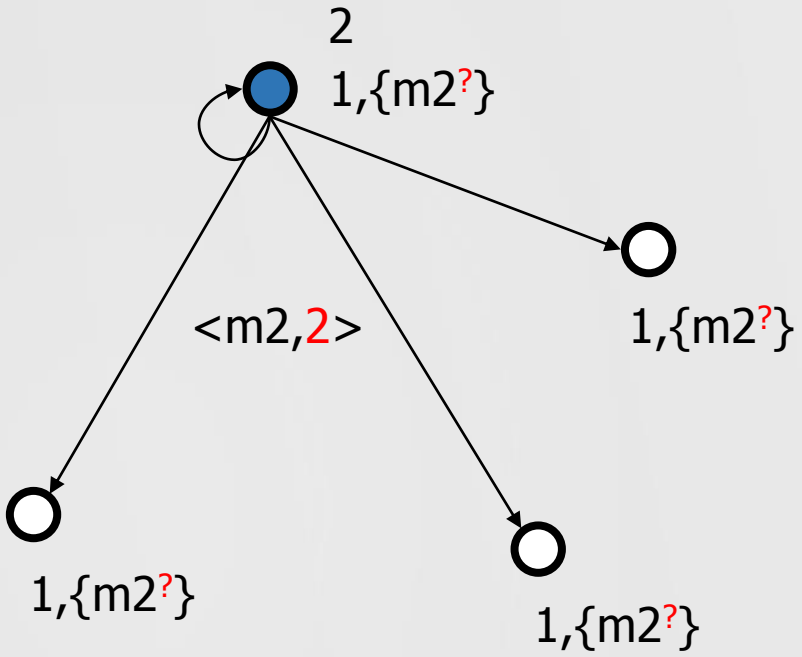
Πρωτόκολλο με συντονιστή (10)



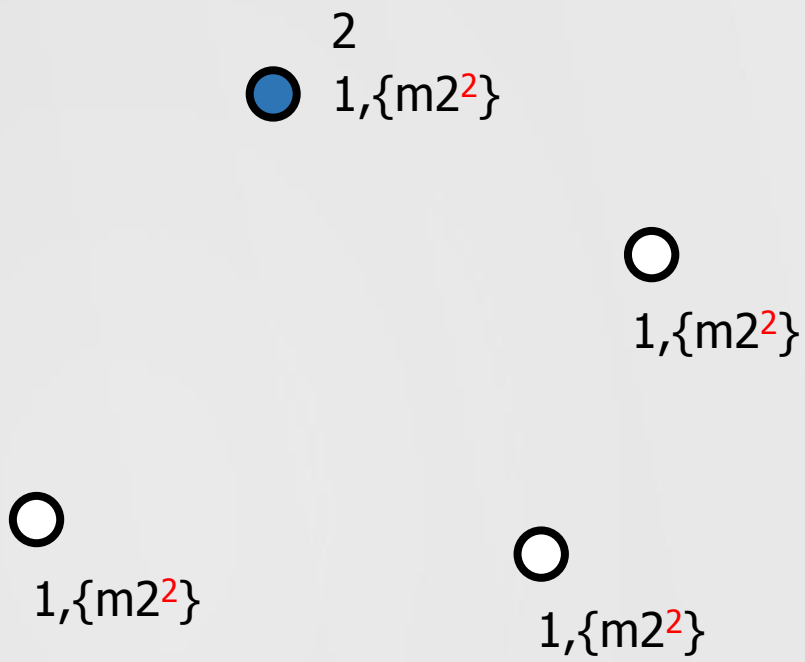
Πρωτόκολλο με συντονιστή (11)



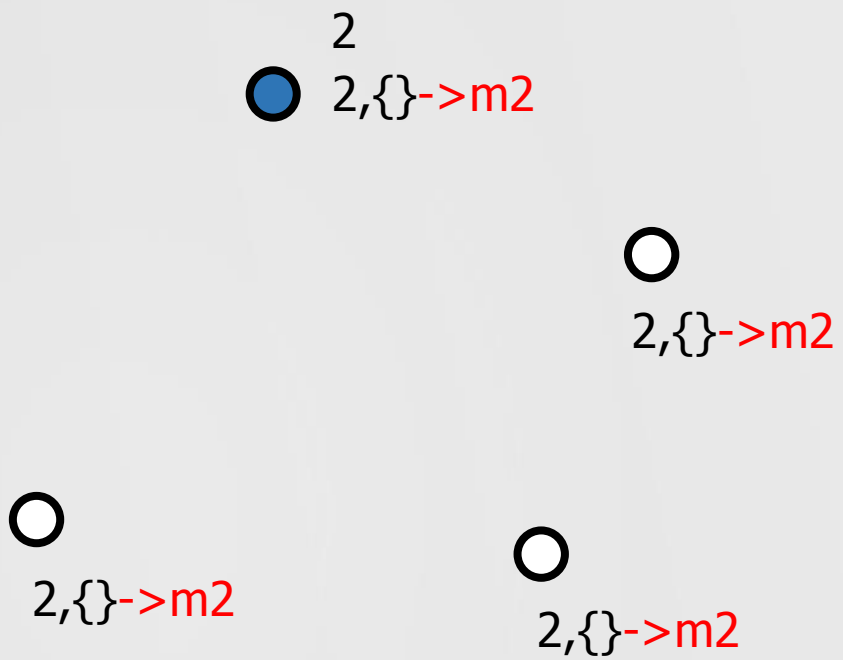
Πρωτόκολλο με συντονιστή (12)



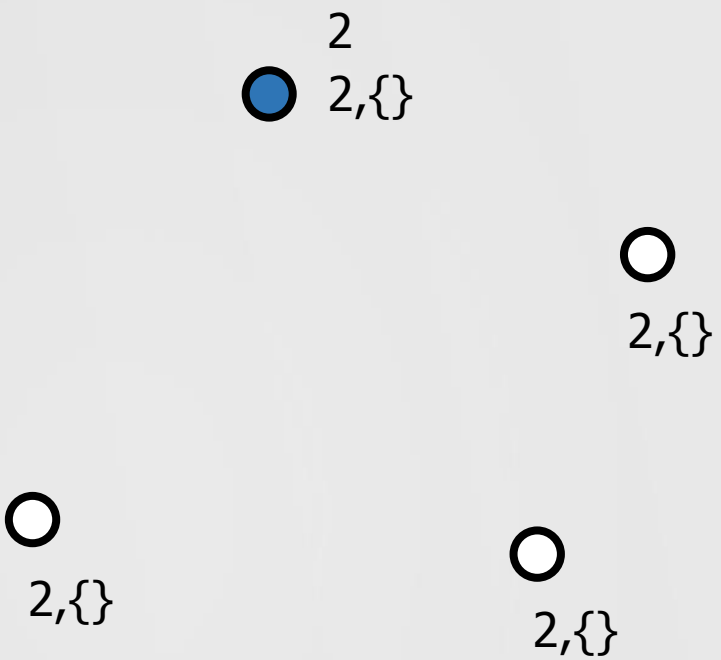
Πρωτόκολλο με συντονιστή (13)



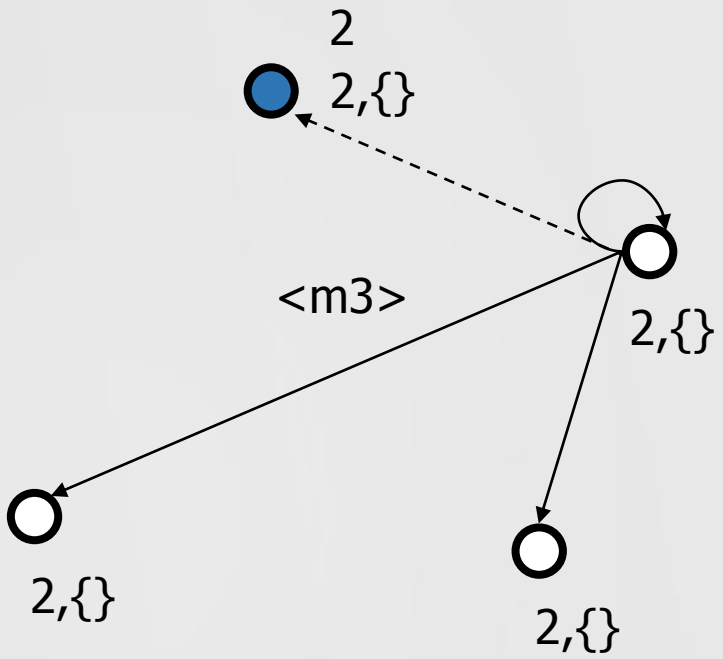
Πρωτόκολλο με συντονιστή (14)



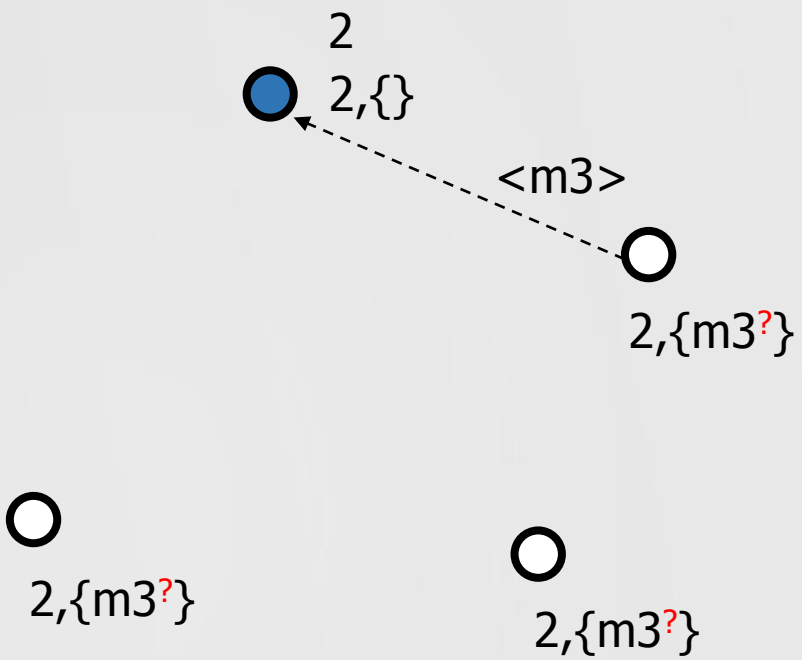
Πρωτόκολλο με συντονιστή (15)



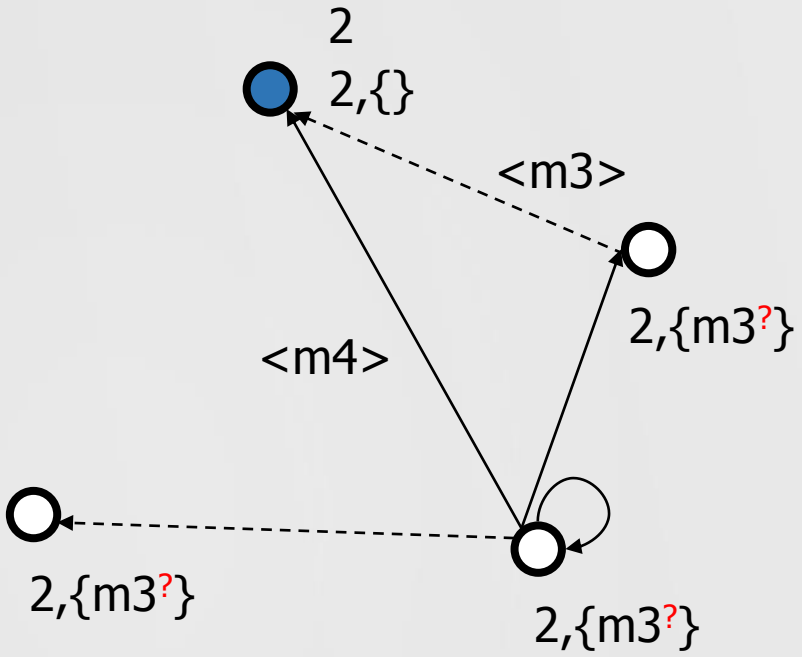
Πρωτόκολλο με συντονιστή (16)



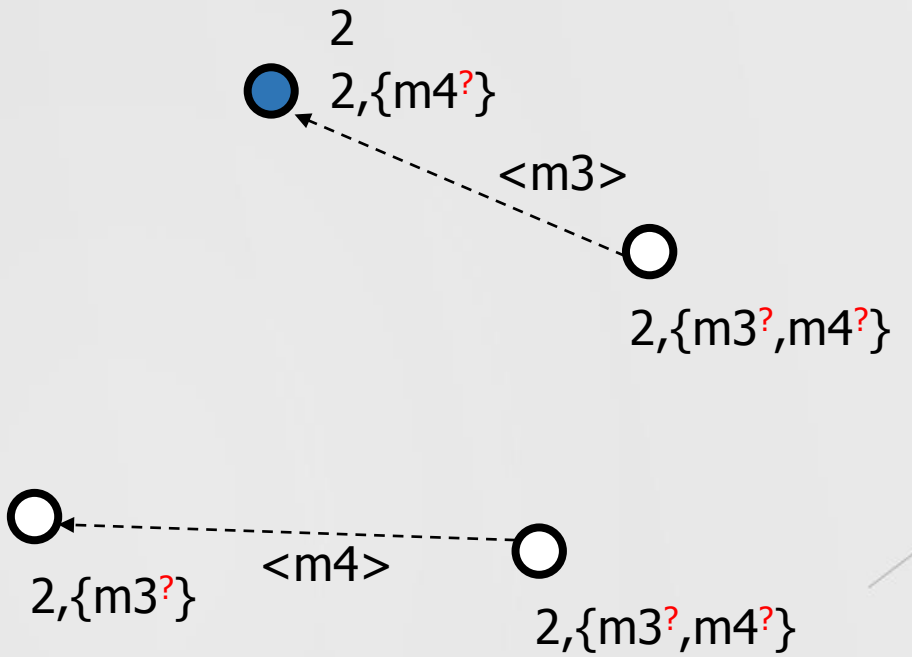
Πρωτόκολλο με συντονιστή (17)



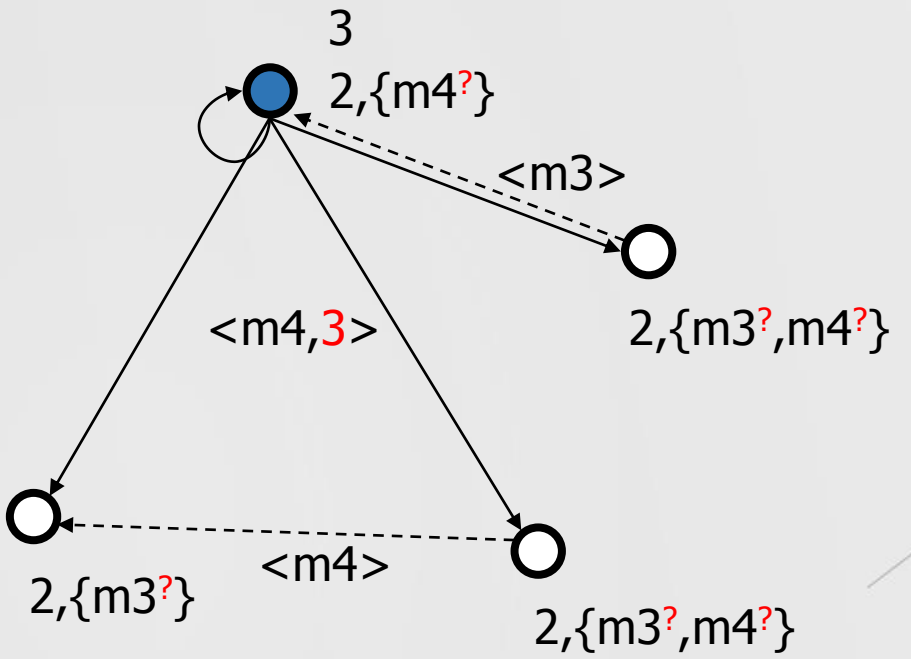
Πρωτόκολλο με συντονιστή (18)



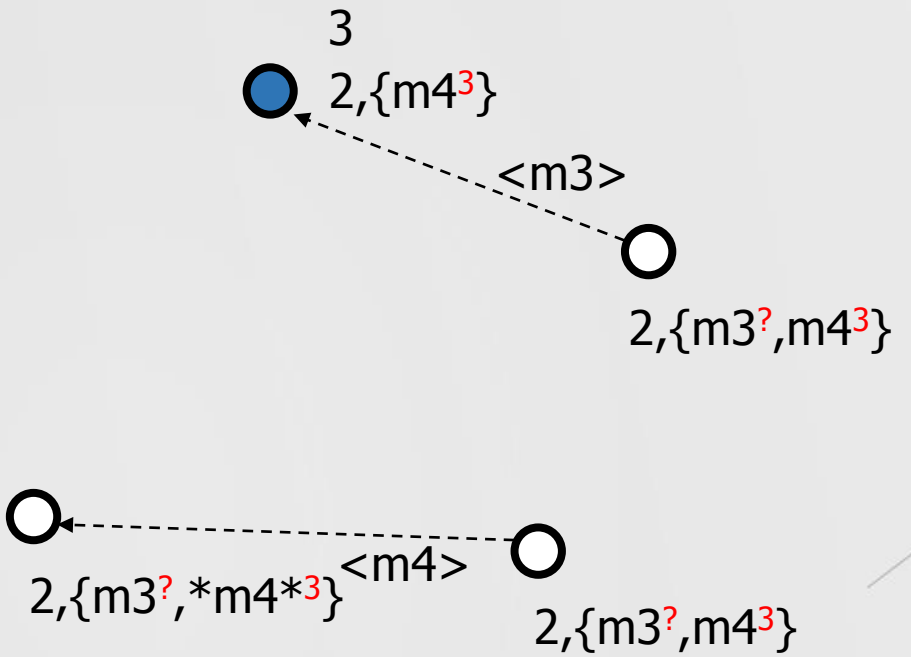
Πρωτόκολλο με συντονιστή (19)



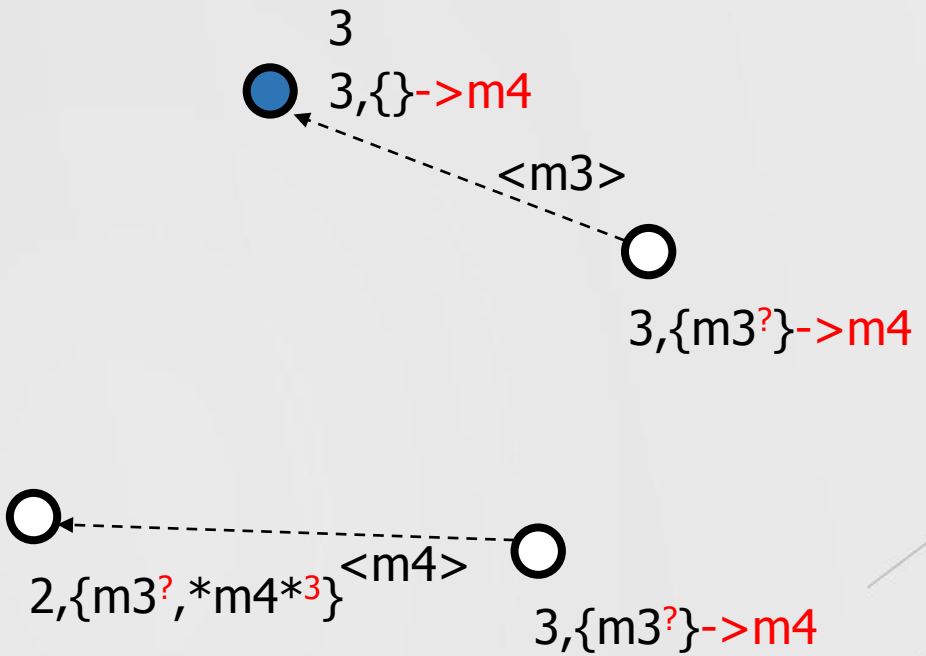
Πρωτόκολλο με συντονιστή (20)



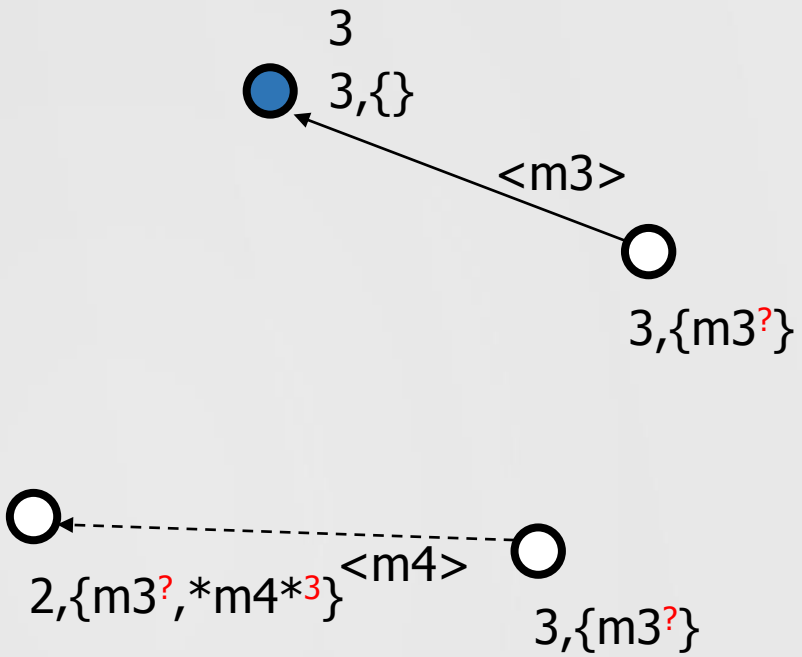
Πρωτόκολλο με συντονιστή (21)



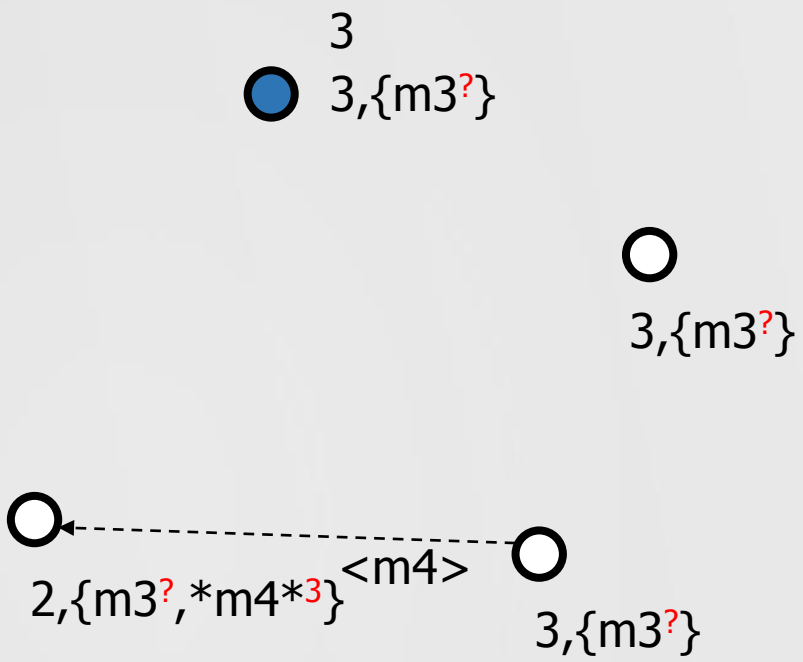
Πρωτόκολλο με συντονιστή (22)



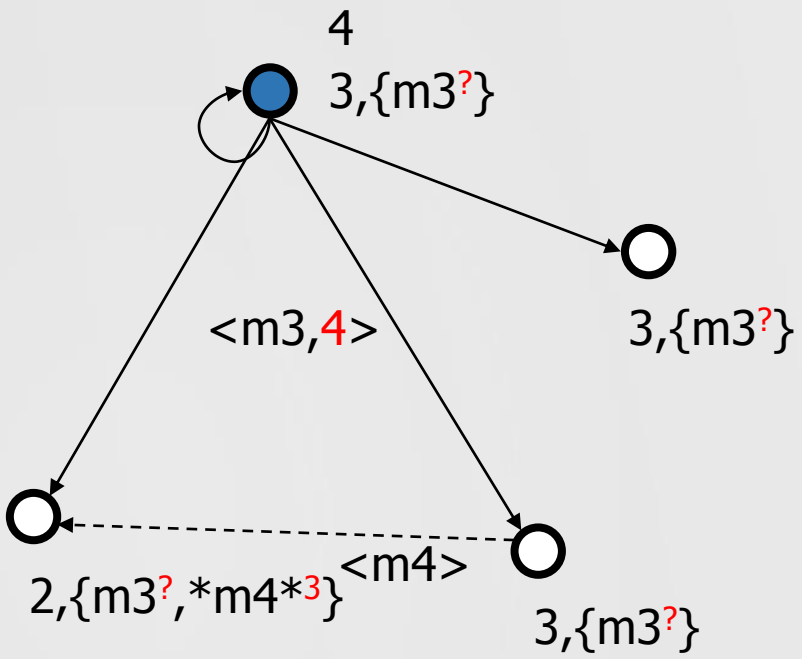
Πρωτόκολλο με συντονιστή (23)



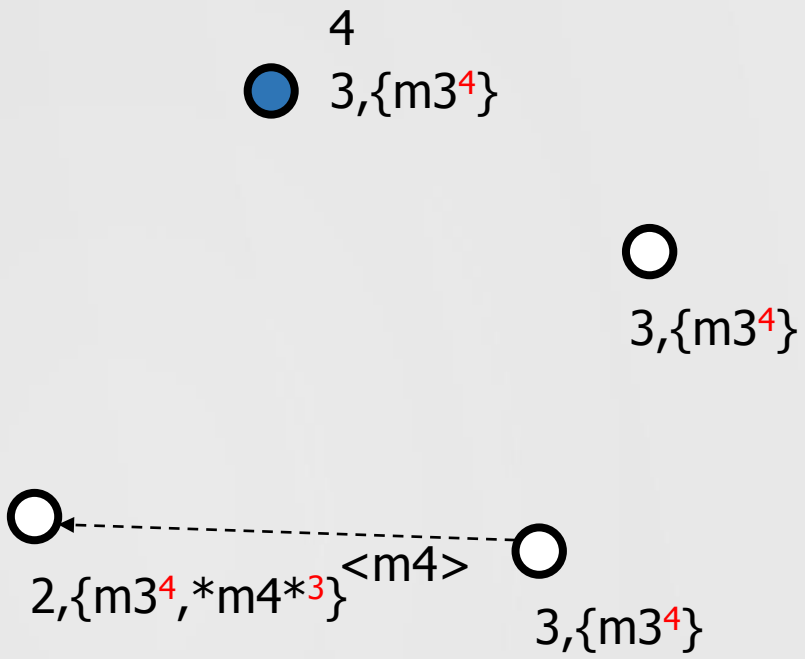
Πρωτόκολλο με συντονιστή (24)



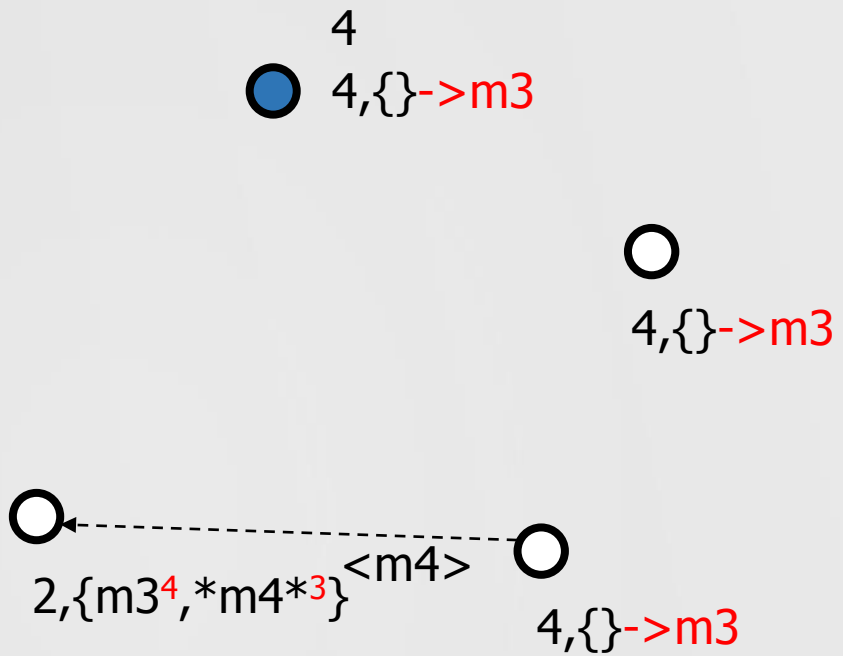
Πρωτόκολλο με συντονιστή (25)



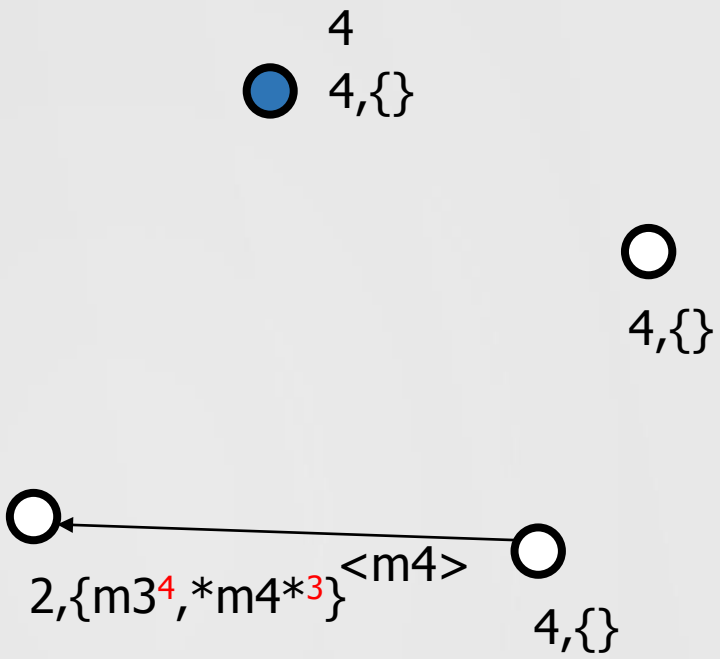
Πρωτόκολλο με συντονιστή (26)



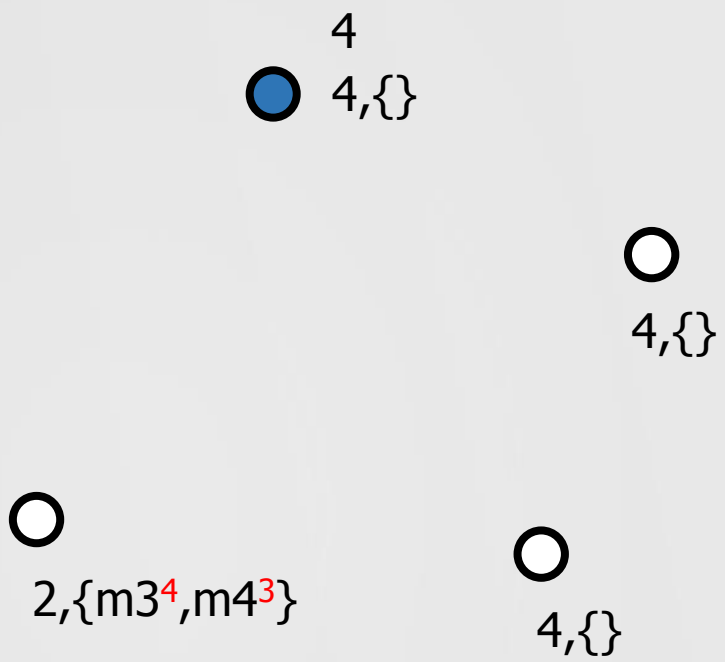
Πρωτόκολλο με συντονιστή (27)



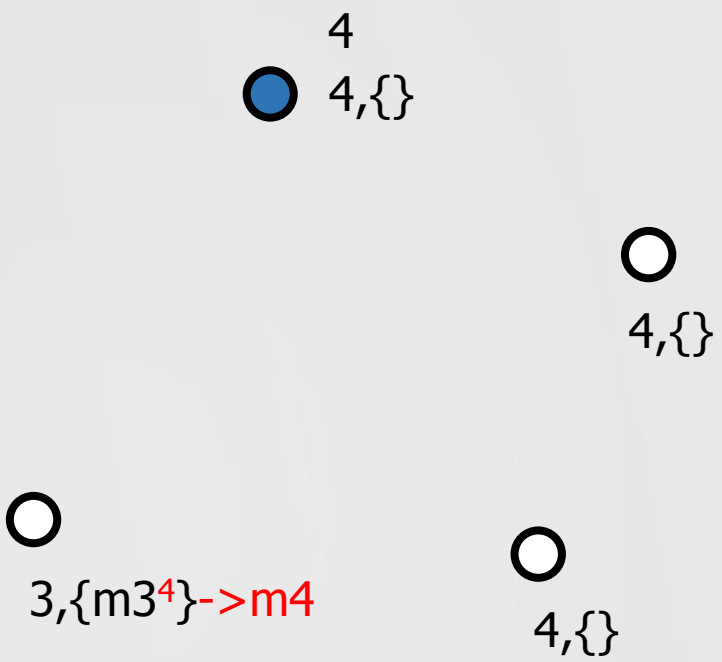
Πρωτόκολλο με συντονιστή (28)



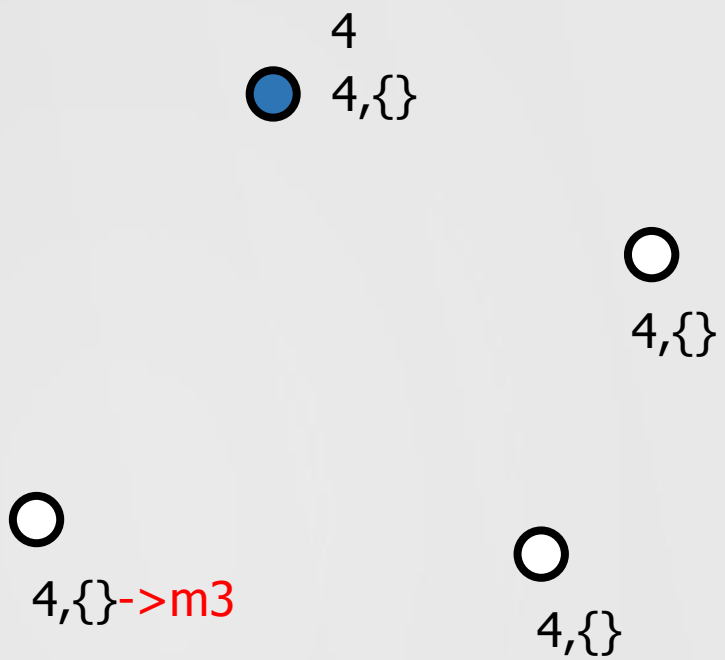
Πρωτόκολλο με συντονιστή (29)



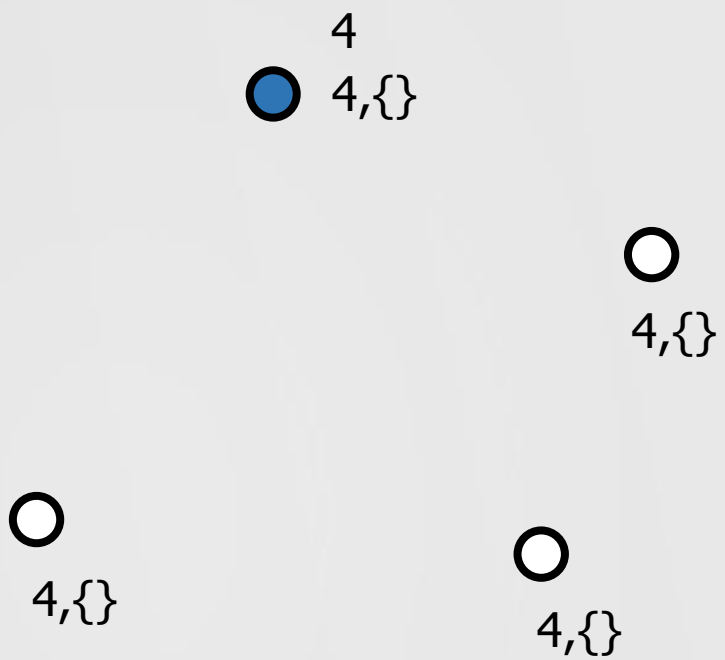
Πρωτόκολλο με συντονιστή (30)



Πρωτόκολλο με συντονιστή (31)



Πρωτόκολλο με συντονιστή (32)



Βλάβες – βελτιστοποιήσεις

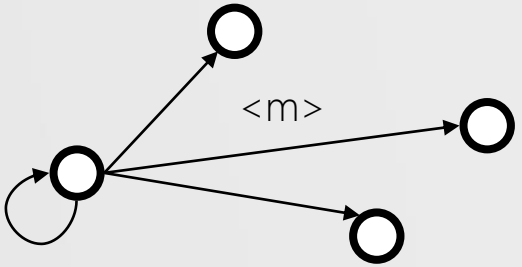
- Αν ο συντονιστής παρουσιάσει βλάβη, πρέπει να επιλεγεί νέος που θα συνεχίσει την έκδοση σειριακών αριθμών από εκεί που σταμάτησε ο προηγούμενος.
- Δεν είναι σίγουρο ότι όλοι έχουν ήδη λάβει ή θα λάβουν τον τελευταίο σειριακό αριθμό που έστειλε ο παλιός συντονιστής.
- Δεν είναι σίγουρο ότι όλοι έχουν λάβει ή θα λάβουν το αντίστοιχο μήνυμα εφαρμογής.
- Μπορεί να γίνουν διάφορες βελτιστοποιήσεις.
- φυσική πολυεκπομπή, σχήματα επιβεβαίωσης, ...

Συμμετρικό πρωτόκολλο

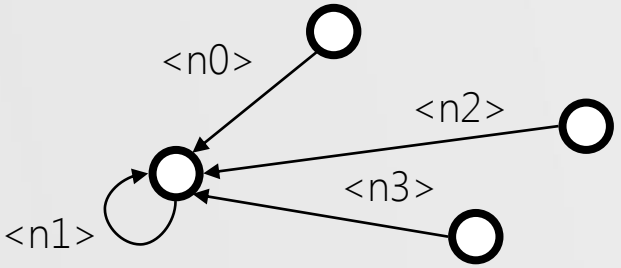
- Κάθε μήνυμα στέλνεται σε όλα τα μέλη της ομάδας.
- Όταν μια διεργασία λάβει ένα μήνυμα, το τοποθετεί σε αποθήκη, αυξάνει έναν προτεινόμενο σειριακό αριθμό παράδοσης που στέλνει στον αποστολέα και περιμένει να λάβει τον «οριστικό» σειριακό αριθμό παράδοσης.
- Ο αποστολέας περιμένει προτάσεις από όλες τις διεργασίες, επιλέγει ως οριστικό σειριακό αριθμό την μεγαλύτερη πρόταση και την στέλνει σε όλα τα μέλη.
- Κάθε διεργασία παραδίδει τα μηνύματα με βάση τον οριστικό σειριακό που στέλνει ο αποστολέας (ανανεώνοντας την τοπική τιμή αντίστοιχα).

Συμμετρικό πρωτόκολλο

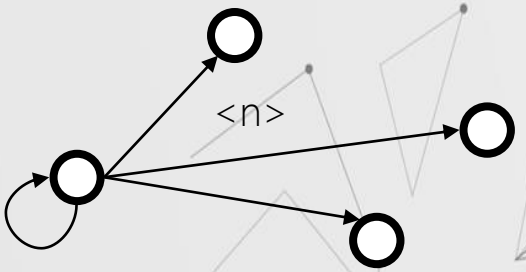
1:
το μήνυμα στέλνεται σε όλα
τα μέλη της ομάδας



2:
κάθε μέλος προτείνει έναν
σειριακό αριθμό παράδοσης
(που αυξάνεται κατάλληλα)



3:
ο αποστολέας επιλέγει την
μεγαλύτερη τιμή και την στέλνει
σε όλα τα μέλη



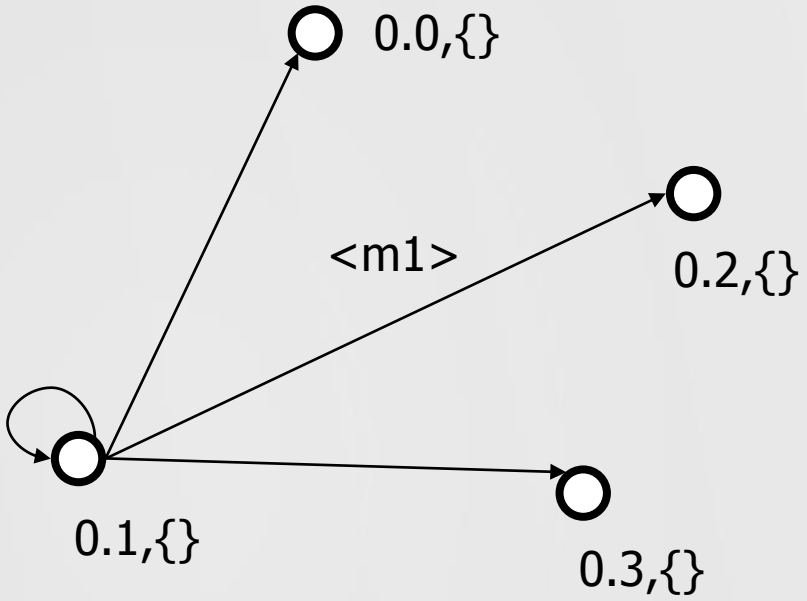
4:
κάθε μέλος παραδίδει τα μηνύματα
που λαμβάνει σύμφωνα με τον
οριστικό σειριακό αριθμό

Σειριακοί αριθμοί

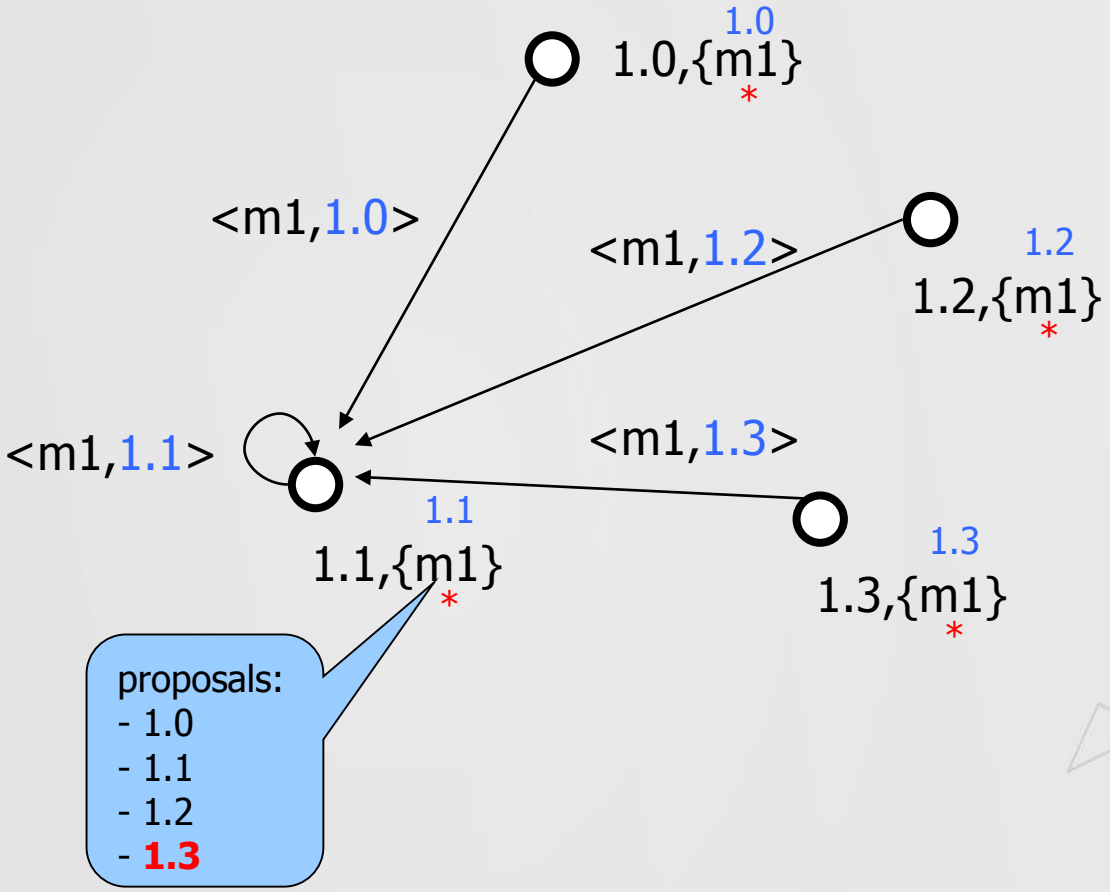
- Ένα μήνυμα δεν μπορεί να παραδοθεί στην εφαρμογή χωρίς να το έχουν λάβει (και να έχουν προτείνει έναν σειριακό αριθμό παράδοσης) όλα τα μέλη της ομάδας.
- Αν μια διεργασία προτείνει σειριακό αριθμό, ξέρει ότι αυτό το μήνυμα δεν θα λάβει μικρότερο οριστικό σειριακό αριθμό – μπορεί όμως να λάβει μεγαλύτερο.
- Σειριακοί αριθμοί της μορφής $(k.P)$, ώστε να μην μπορούν δύο μηνύματα να λάβουν τον ίδιο αριθμό.
- Σε περίπτωση «ισότητας» ανάμεσα σε $(k.P1)$ και $(k.P2)$, γίνεται σύγκριση των αναγνωριστικών $P1$ και $P2$.

Συμμετρικό πρωτόκολλο (1)

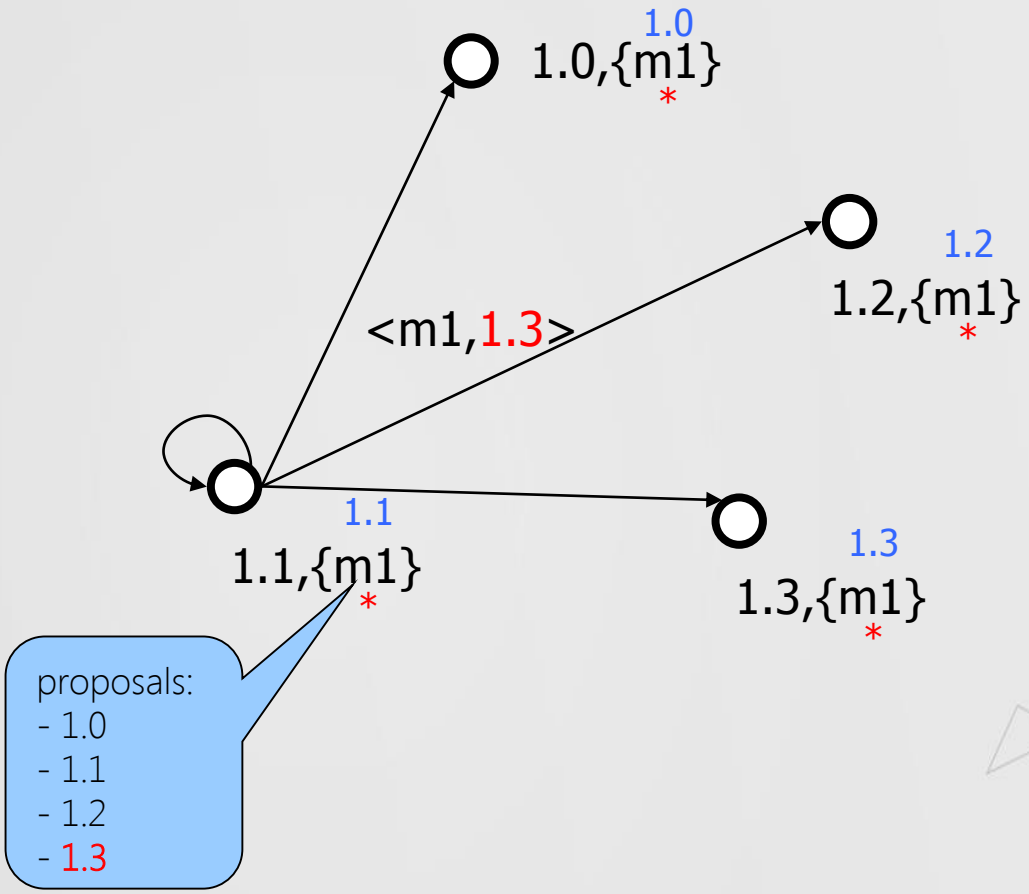
Συμμετρικό πρωτόκολλο (2)



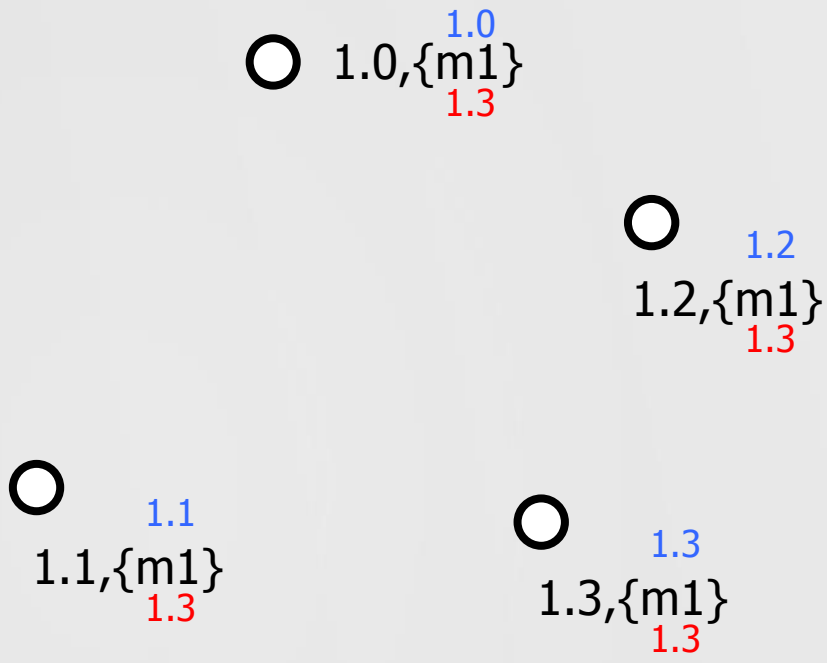
Συμμετρικό πρωτόκολλο (3)



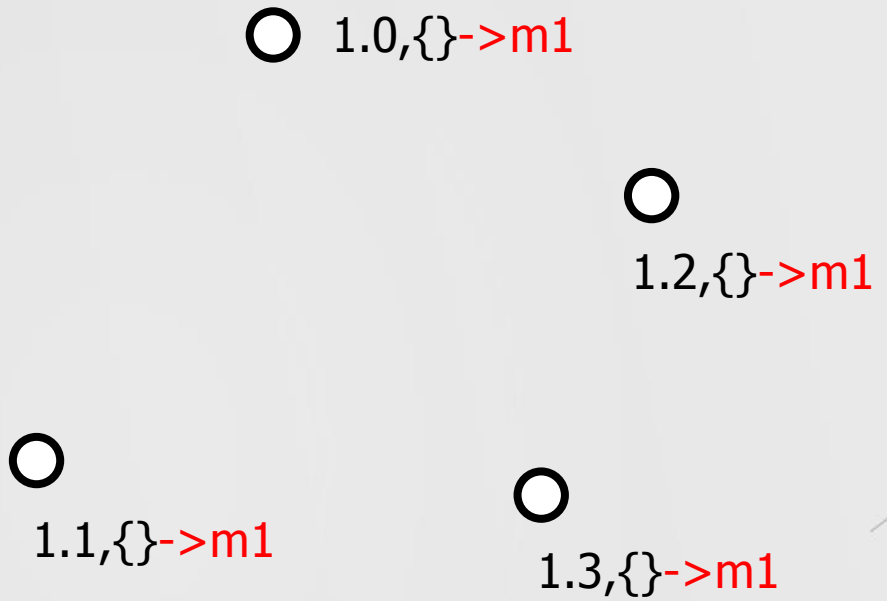
Συμμετρικό πρωτόκολλο (4)



Συμμετρικό πρωτόκολλο (5)

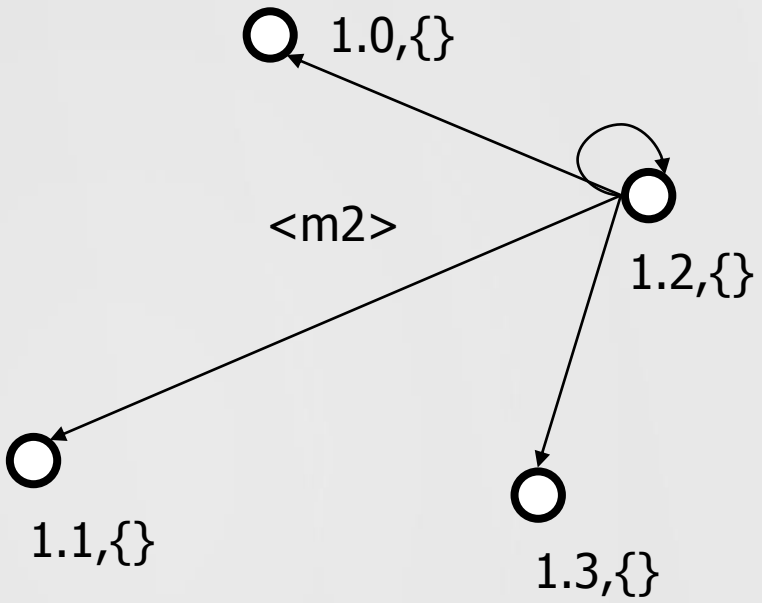


Συμμετρικό πρωτόκολλο (6)

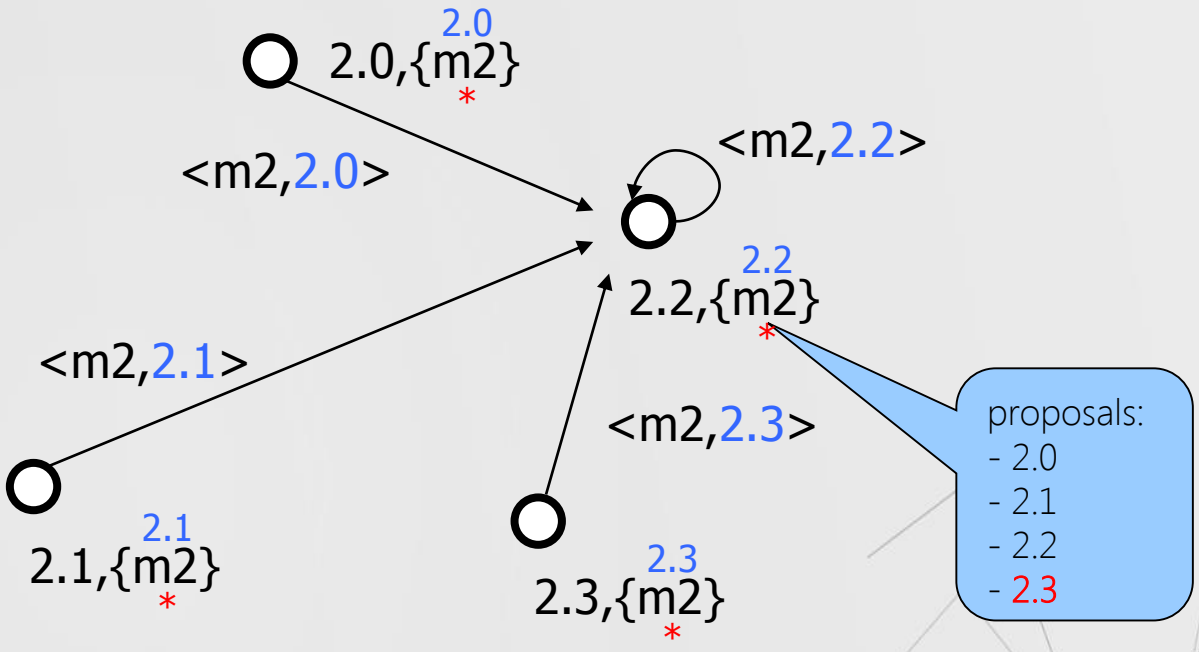


Συμμετρικό πρωτόκολλο (7)

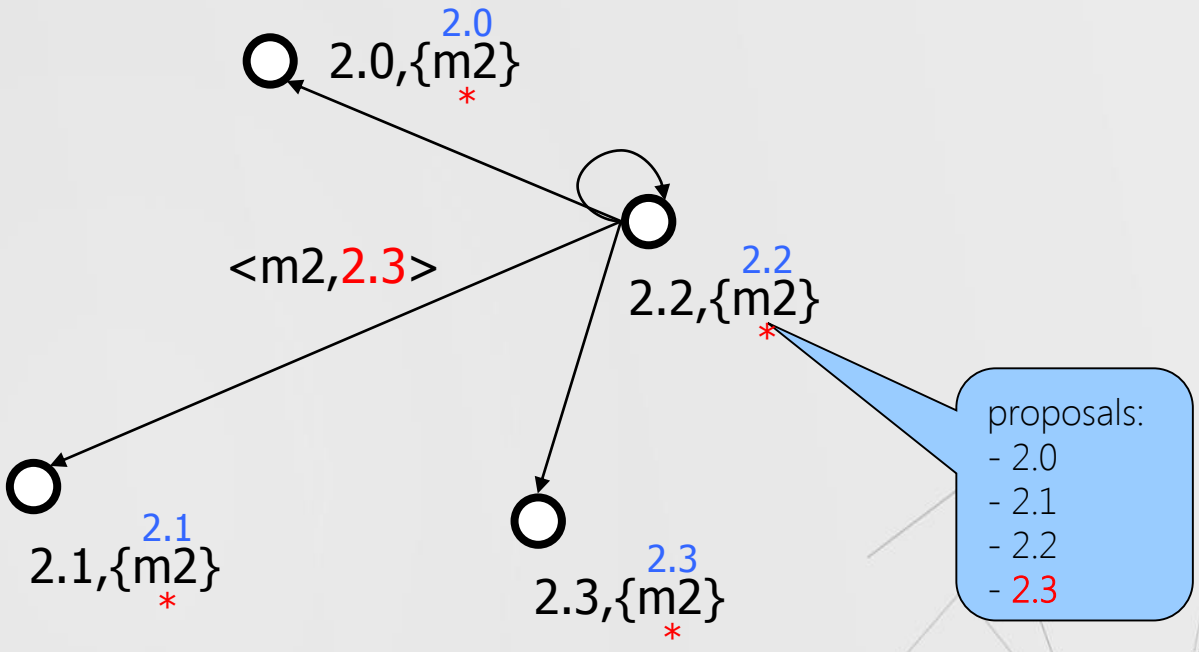
Συμμετρικό πρωτόκολλο (8)



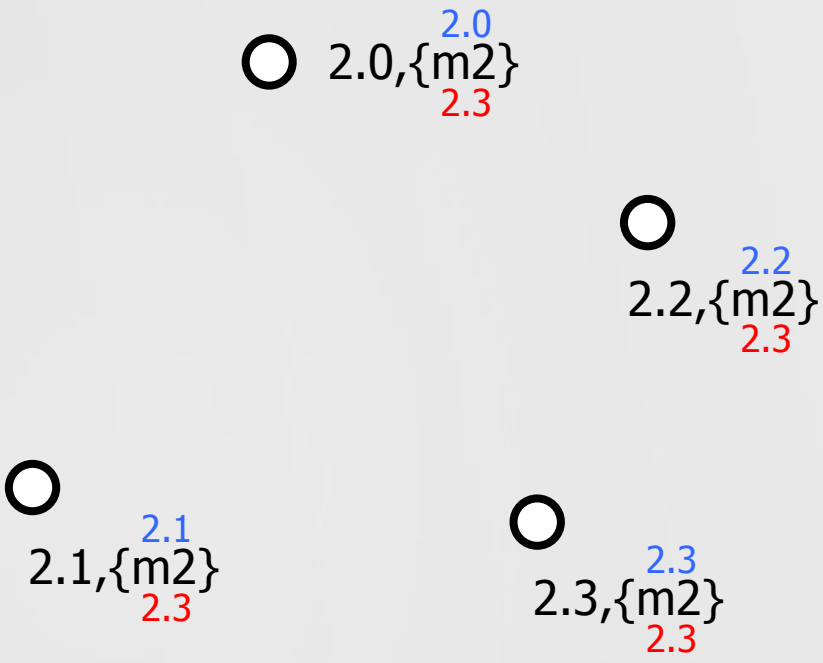
Συμμετρικό πρωτόκολλο (9)



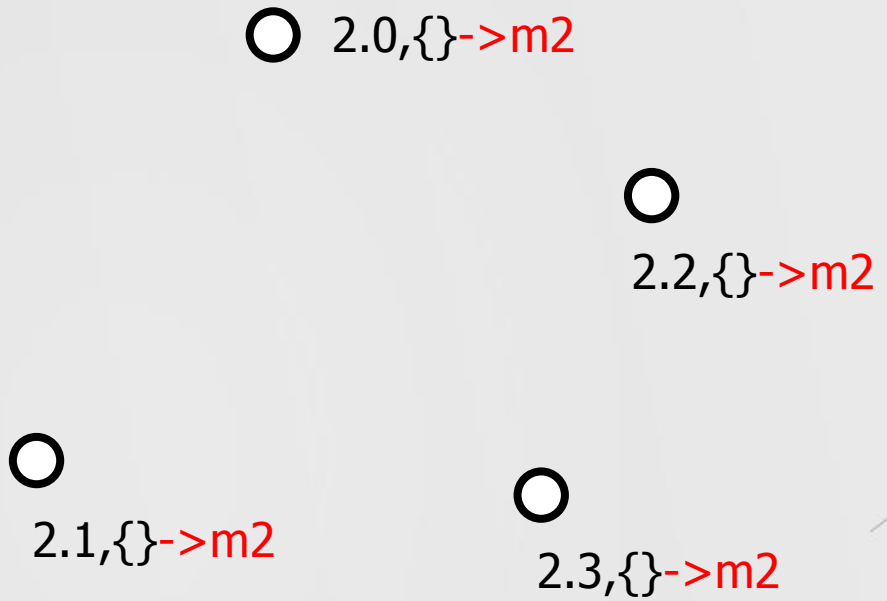
Συμμετρικό πρωτόκολλο (10)



Συμμετρικό πρωτόκολλο (11)

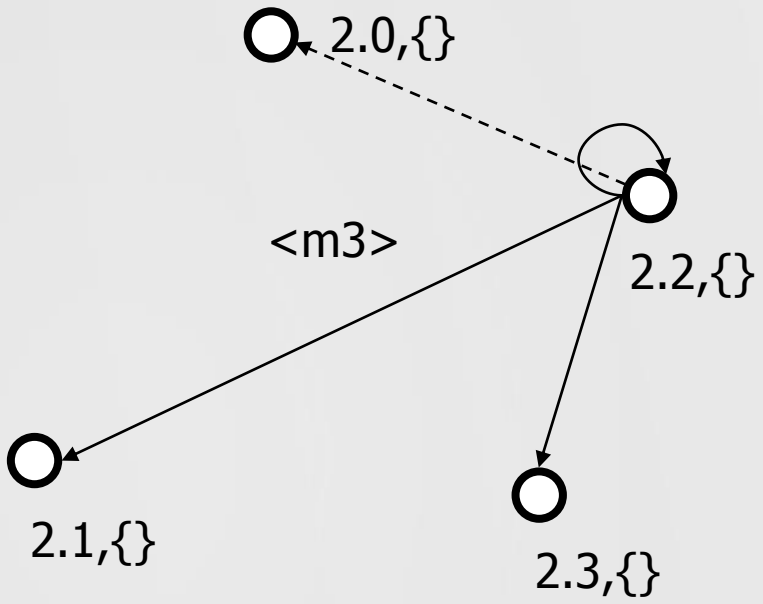


Συμμετρικό πρωτόκολλο (12)

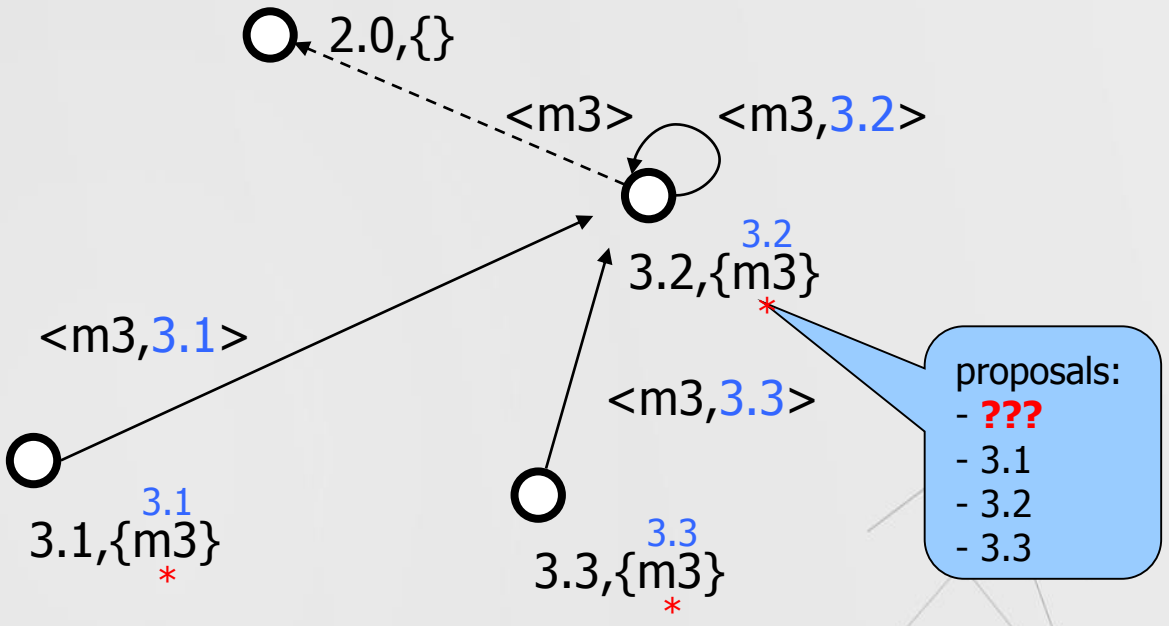


Συμμετρικό πρωτόκολλο (13)

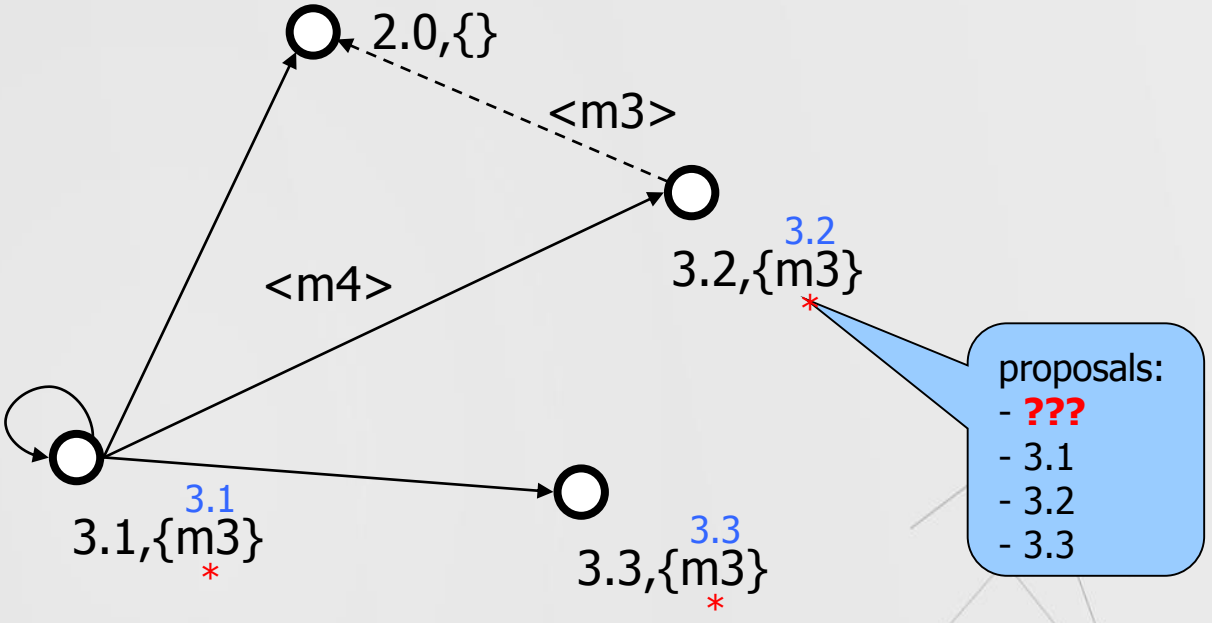
Συμμετρικό πρωτόκολλο (14)



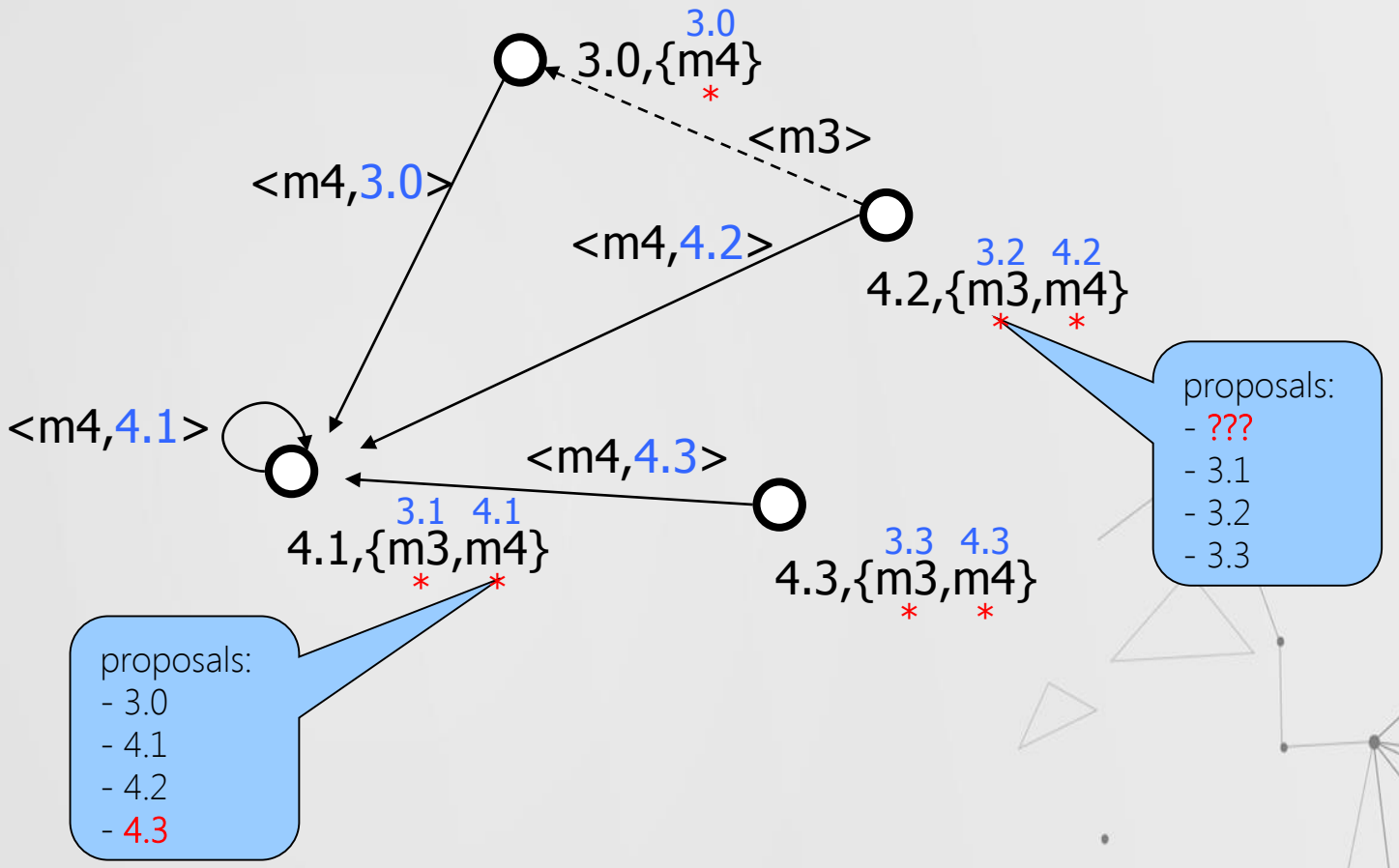
Συμμετρικό πρωτόκολλο (15)



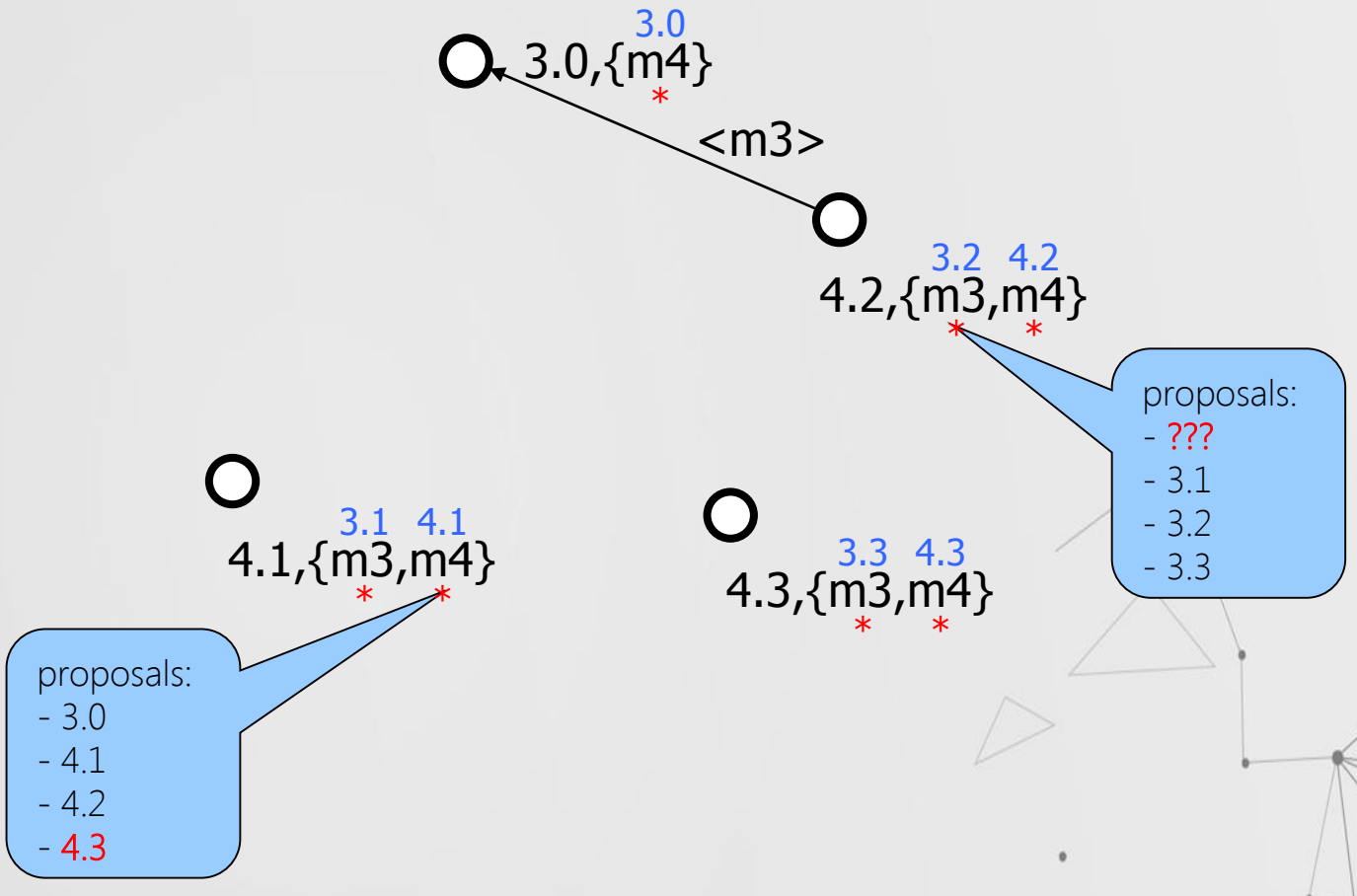
Συμμετρικό πρωτόκολλο (16)



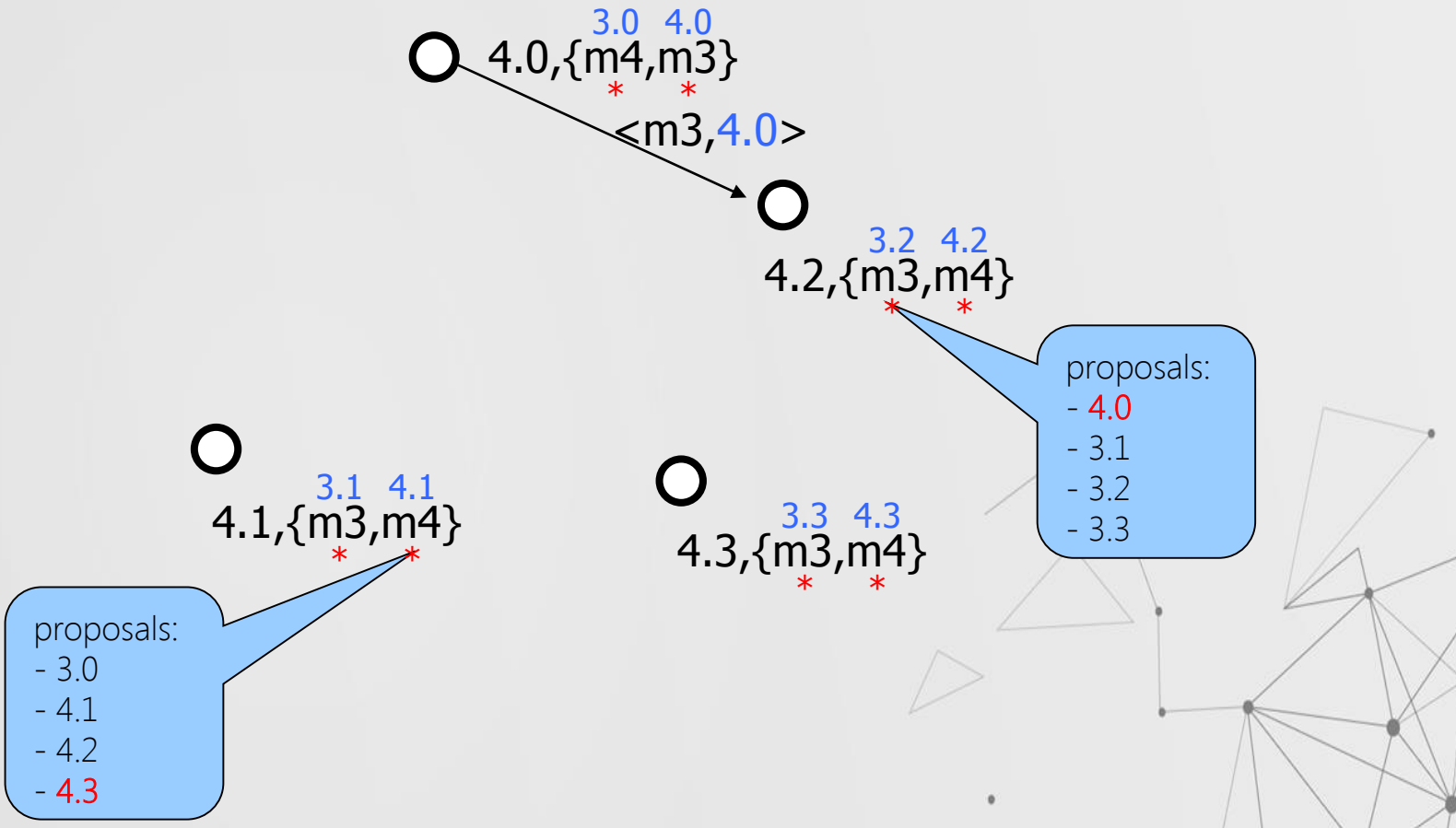
Συμμετρικό πρωτόκολλο (17)



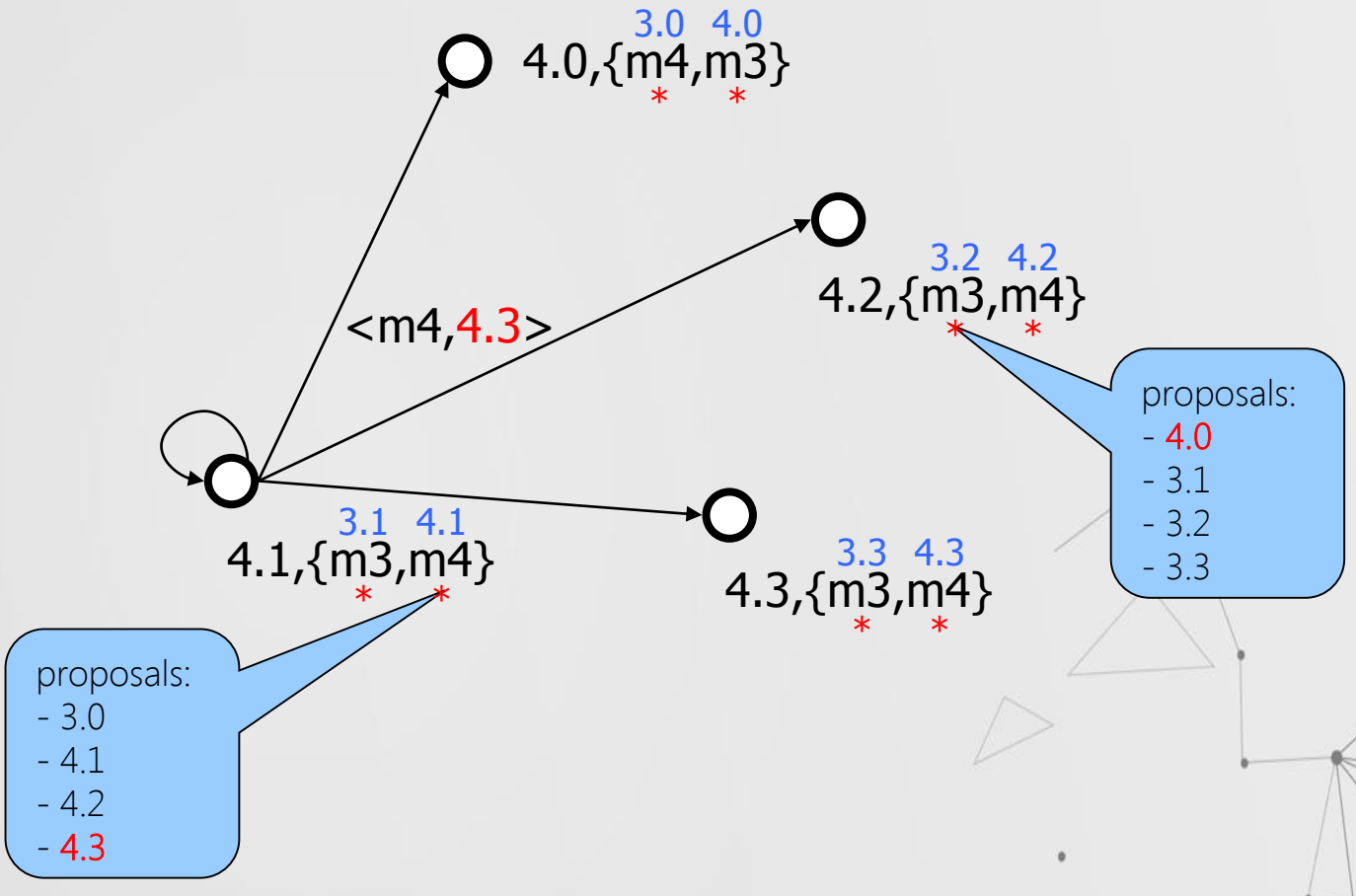
Συμμετρικό πρωτόκολλο (18)



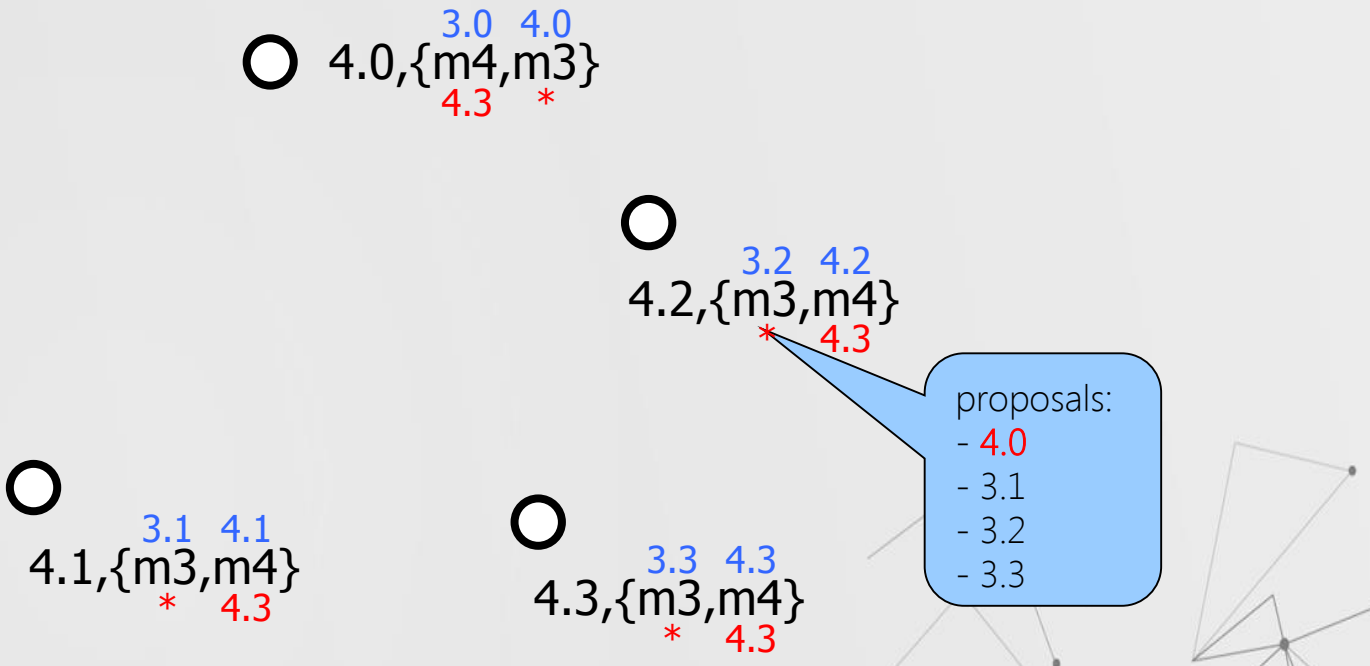
Συμμετρικό πρωτόκολλο (19)



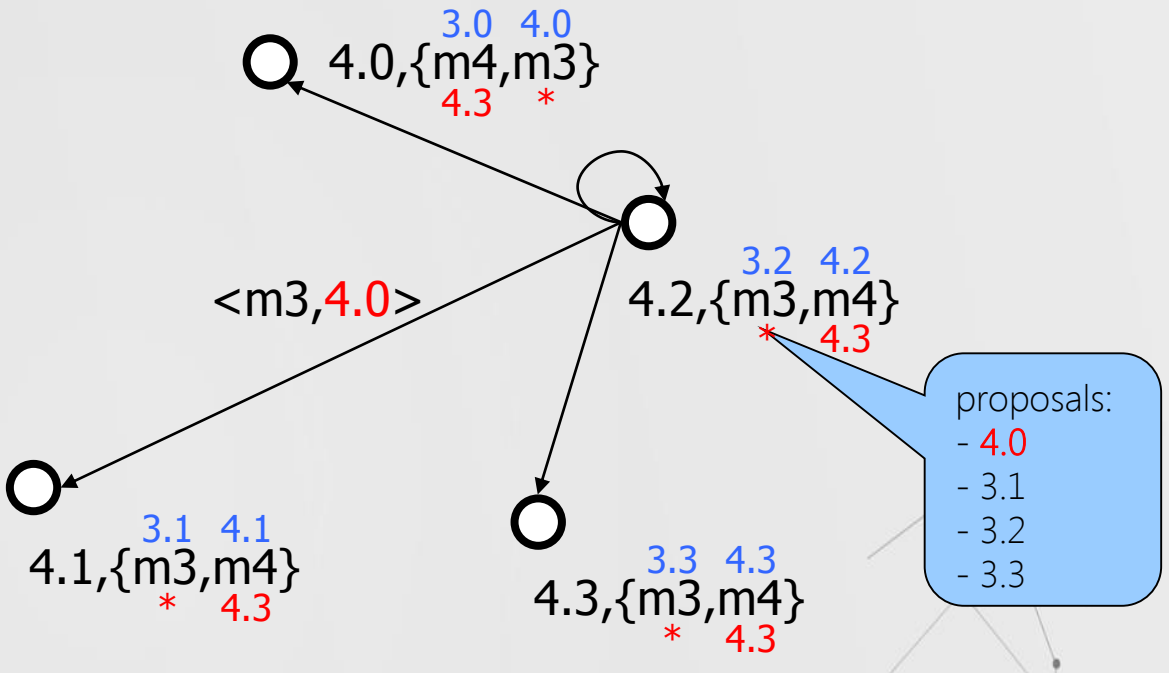
Συμμετρικό πρωτόκολλο (20)



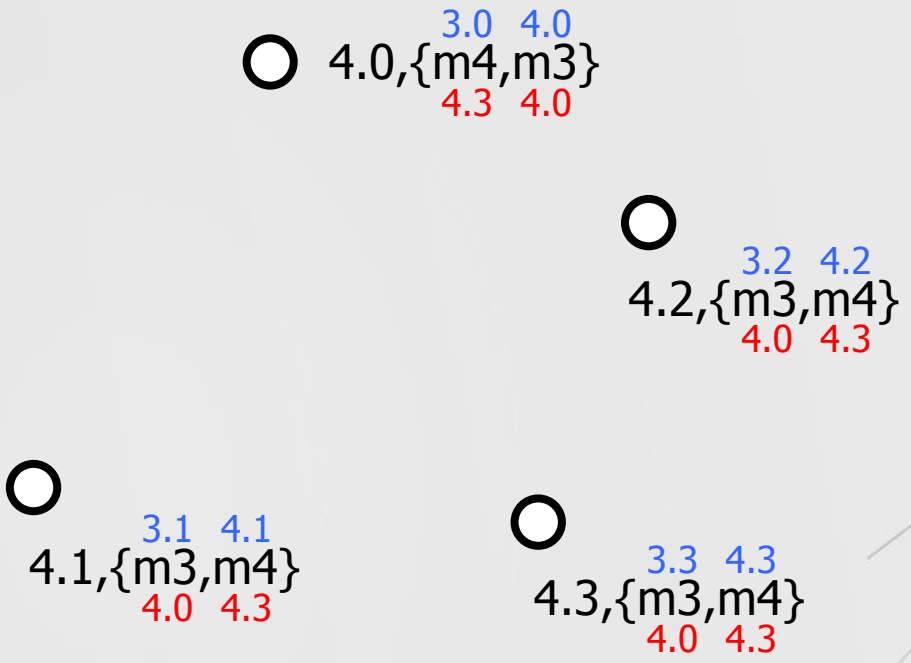
Συμμετρικό πρωτόκολλο (21)



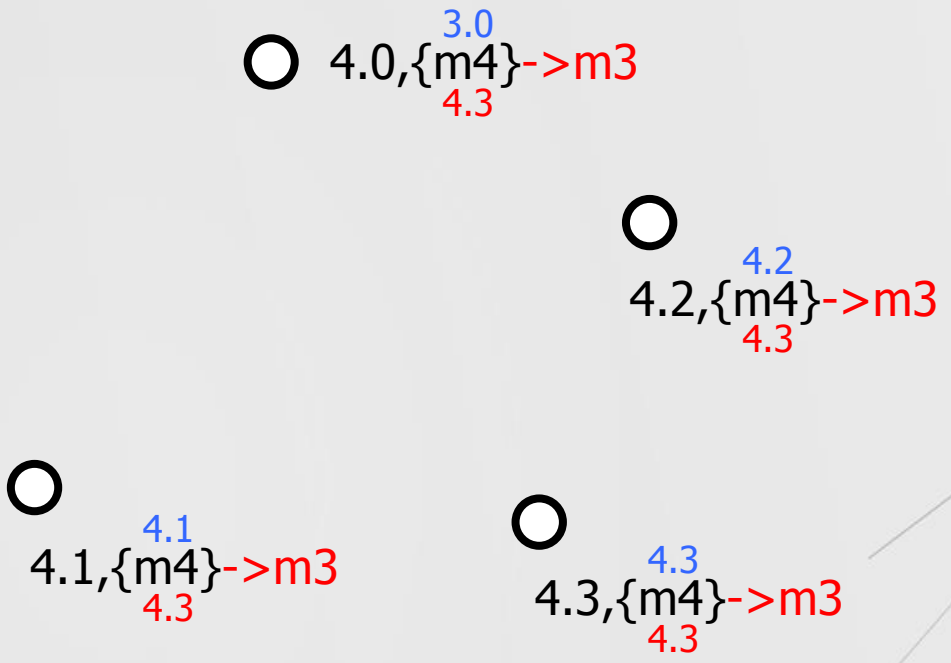
Συμμετρικό πρωτόκολλο (21)



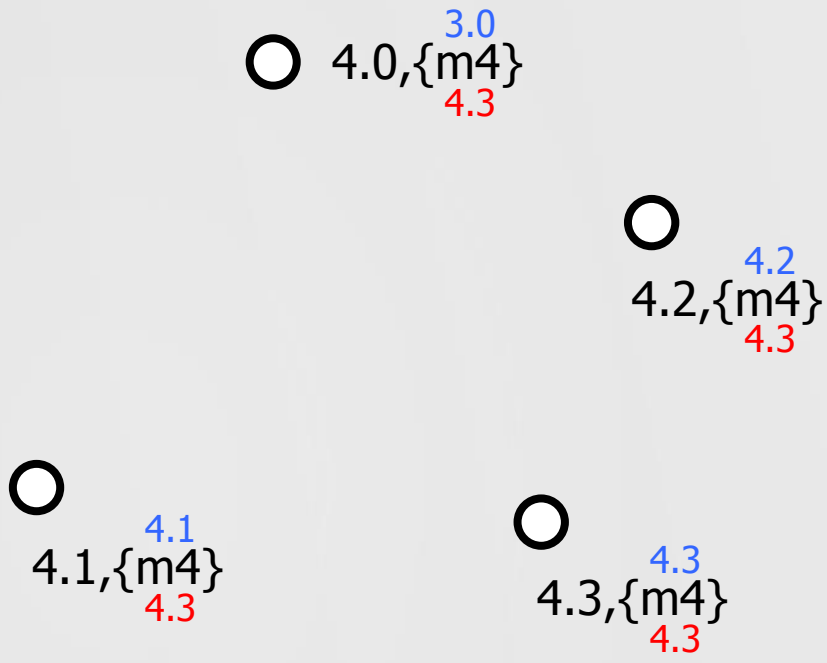
Συμμετρικό πρωτόκολλο (22)



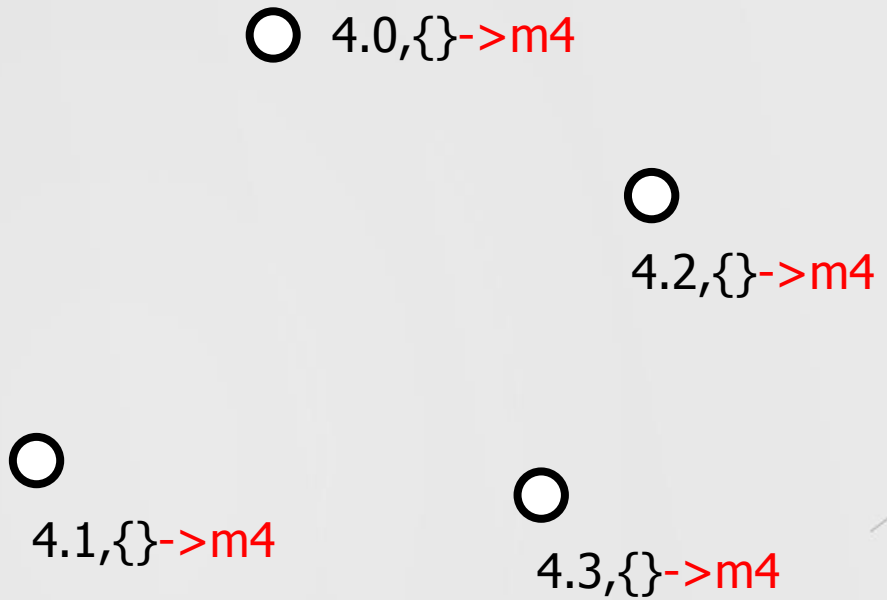
Συμμετρικό πρωτόκολλο (23)



Συμμετρικό πρωτόκολλο (24)



Συμμετρικό πρωτόκολλο (25)



Συμμετρικό πρωτόκολλο (26)

Βλάβες – βελτιστοποιήσεις

- Αν μια διεργασία παρουσιάσει βλάβη, μπορεί να αφήσει «ορφανό» κάποιο δικό της μήνυμα – χωρίς να έχει στείλει (σε όλους) τον οριστικό σειριακό αριθμό του.
- Απαιτείται συνεργασία των υπολοίπων διεργασιών για την κατάλληλη «ολοκλήρωση» της μετάδοσης και παράδοσης του μηνύματος.
- Σε σχέση με την κεντρικοποιημένη μέθοδο με συντονιστή, απαιτούνται περισσότερα μηνύματα.
- 2 μηνύματα RM, και N μηνύματα unicast.
- Μπορεί να γίνουν διάφορες βελτιστοποιήσεις.
  - φυσική πολυεκπομπή, σχήματα επιβεβαίωσης, …

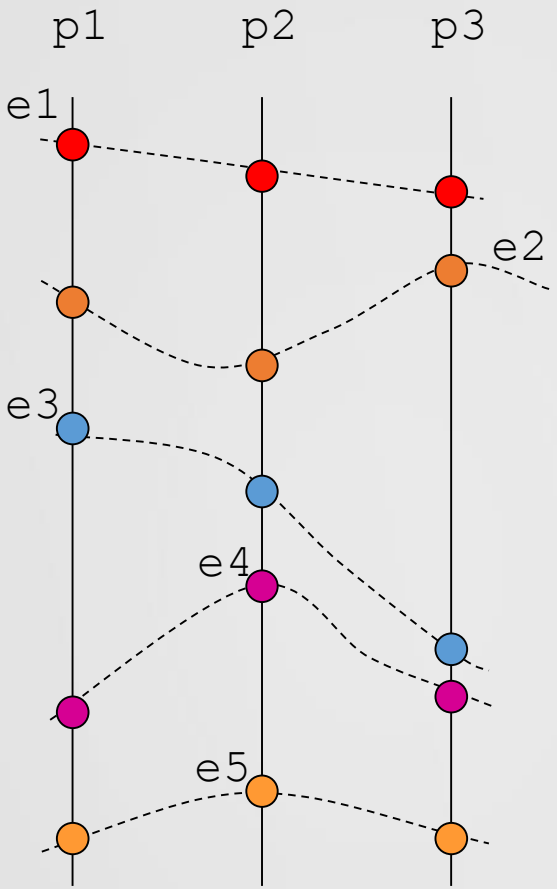
Πρωτόκολλα total-causal multicast

- Ένα πρωτόκολλο total multicast δεν εγγυάται (απαραίτητα) παράδοση με αιτιολογική σειρά.
- Αν έχει ήδη παραδοθεί το m1 και στη συνέχεια σταλεί το m2, λόγω total multicast, το m2 όντως θα λάβει μεγαλύτερο αριθμό σειράς και θα παραδοθεί μετά το m1.
- Όμως, αν η ίδια διεργασία στείλει πρώτα το m1 και μετά το m2, δεν είναι εγγυημένο ότι το m2 θα λάβει μεγαλύτερο αριθμό σειράς από το m1.
- Τα προηγούμενα πρωτόκολλα χρειάζονται κατάλληλες επεκτάσεις FIFO στην μετάδοση μηνυμάτων.

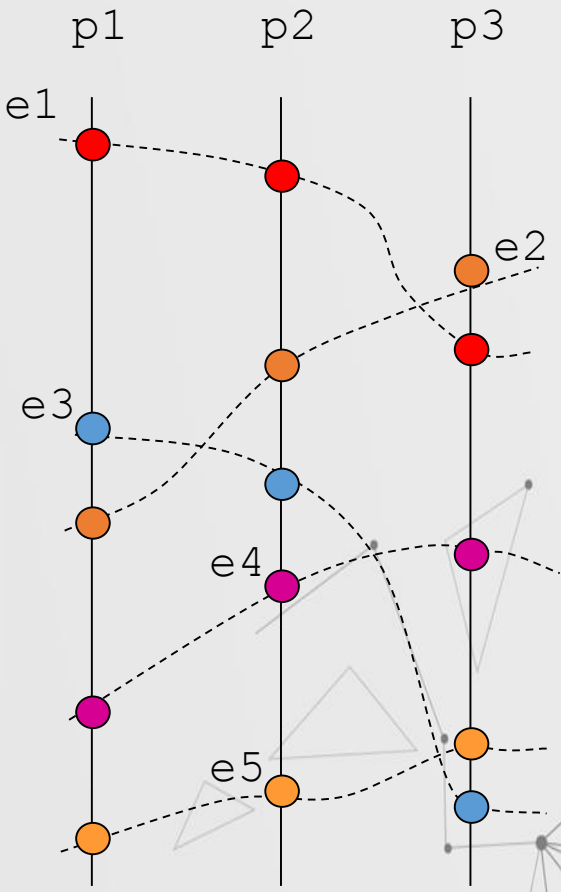
Εικονικός συγχρονισμός με CATOC

- Το causally-and-totally communication επιτυγχάνει έναν «εικονικό συγχρονισμό»,
  - ακόμα και σε ασύγχρονο σύστημα.
- Όλες οι διεργασίες λαμβάνουν όλα τα μηνύματα της ομάδας, με την ίδια & λογικά συνεπή σειρά.
- Οι διεργασίες μπορούν να λάβουν κάποιες αποφάσεις αποκεντρωμένα ή/και να πραγματοποιήσουν ένα ντετερμινιστικό υπολογισμό (με βάση την τρέχουσα κατάσταση τους) καταλήγοντας στο ίδιο αποτέλεσμα.
  - έχει προηγηθεί (εκτενής) επικοινωνία σε χαμηλότερο επίπεδο για να «συμφωνηθεί» η σειρά παράδοσης μηνυμάτων.

virtual synchrony

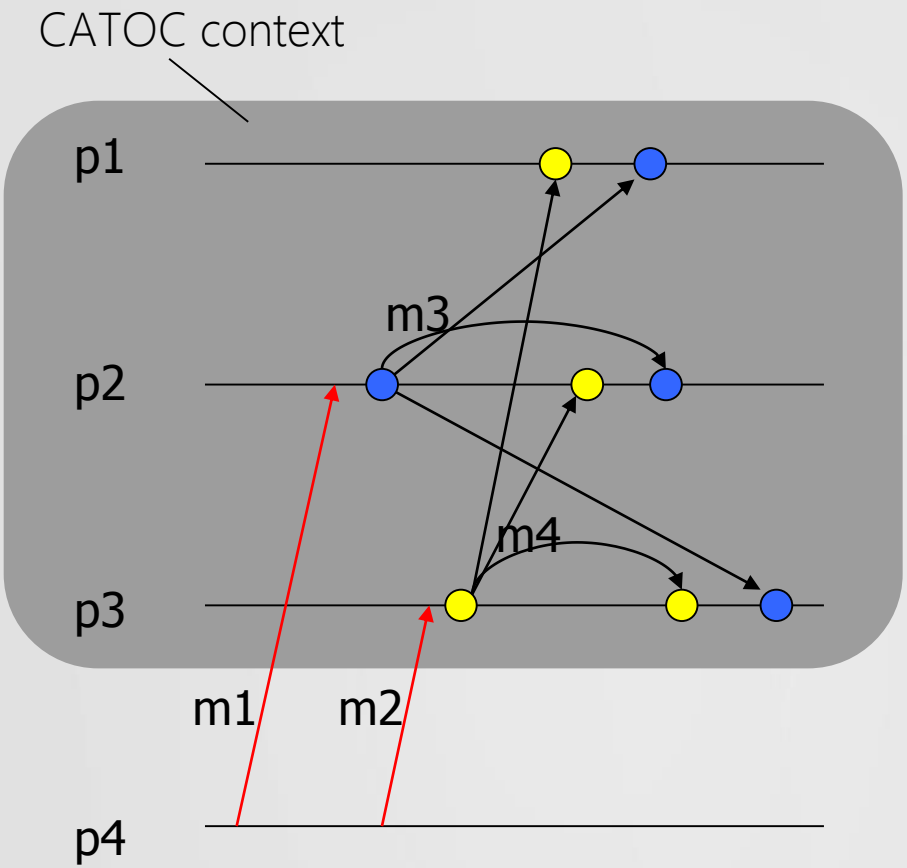


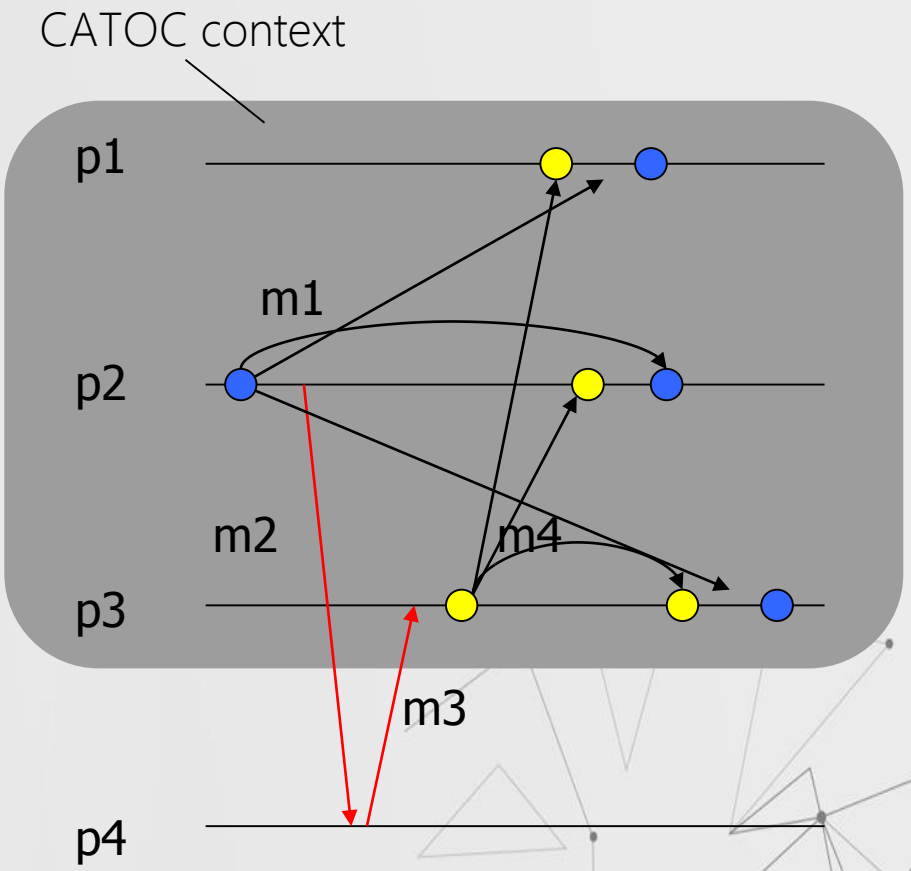
asynchrony



Περιορισμοί CATOC

- Οι μηχανισμοί CATOC παρέχουν συνέπεια μόνο αν
  1. υιοθετούνται από όλα τα τμήματα της εφαρμογής
  2. όλα τα μηνύματα της εφαρμογής στέλνονται μέσα στο ίδιο πλαίσιο συγχρονισμού CATOC.
- Οι λογικοί συσχετισμοί μπορεί να χαθούν όταν χρησιμοποιούνται «κρυφά» κανάλια επικοινωνίας, π.χ., αλληλεπίδραση με διεργασίες που δεν υπάγονται στο ίδιο πλαίσιο συγχρονισμού.
- Ακυρώνεται η συνέπεια που προσφέρει το CATOC.


CATOC context
p1
p2
p3
p4
m3
m4
m1
m2


CATOC context
p1
p2
p3
p4
m1
m4
m2
m3

Λογική συνέπεια σε επίπεδο εφαρμογής

- Η λογική συνέπεια σε επίπεδο εφαρμογής πιθανώς να μην συμβαδίζει / ταυτίζεται με αυτήν του CATOC.
- Μπορεί να χρειάζονται επιπρόσθετοι ή να αρκούν από μόνοι τους διαφορετικοί μηχανισμοί συνέπειας.
  - κλείδωμα, καταγραφή εκδόσεων δεδομένων, δοσοληψίες, κλπ.
- Οι μηχανισμοί συνέπειας (σε ψηλό επίπεδο) μπορεί να κλιμακώνουν καλύτερα από τους μηχανισμούς CATOC.
  - λιγότερα μηνύματα σε επίπεδο δικτύου.