

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ  
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΤΟΜΕΑ «ΘΕΩΡΗΤΙΚΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΘΕΩΡΙΑΣ  
ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΛΕΓΧΟΥ»

# Μοντελοποίηση και έλεγχος εφοδιαστικών αλυσίδων με τη χρήση δικτύων Petri

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μπαλατσού Σπυριδούλα

Επιβλέπων: Παναγιώτης Τζιώνας

Θεσσαλονίκη, Δεκέμβριος 2009

## Σκοπός διπλωματικής

- Η μοντελοποίηση και ο έλεγχος μιας μονάδας παραγωγής με τη βοήθεια των δικτύων Coloured Petri Nets.
- Η ανάλυση, η διερεύνηση και η επεξεργασία των εφοδιαστικών αλυσίδων που αντιστοιχούν στις μονάδες παραγωγής.
- Η μελέτη διαφορετικών σεναρίων που περιγράφουν τρόπους διακίνησης προϊόντων.

## **1. Εφοδιαστική αλυσίδα**

- Ορισμός της εφοδιαστικής αλυσίδας και η χρησιμότητά της

## **2. Petri Nets και Coloured PN**

- Περιγραφή των Petri Nets
- Μετάβαση στα CPN
- Είδη και τρόποι αξιοποίησής τους

### **3. Εφοδιαστική αλυσίδα σχεδιασμένη από CPN**

- Περιγραφή της μοντελοποίησης της εφοδιαστικής αλυσίδας
- Σχεδίαση του δικτύου ώστε να υπολογίζει το κόστος των μελών της
- Είδη συνεχούς-στοχαστικής ανατροφοδότησης

## **4. Ανάλυση δεδομένων**

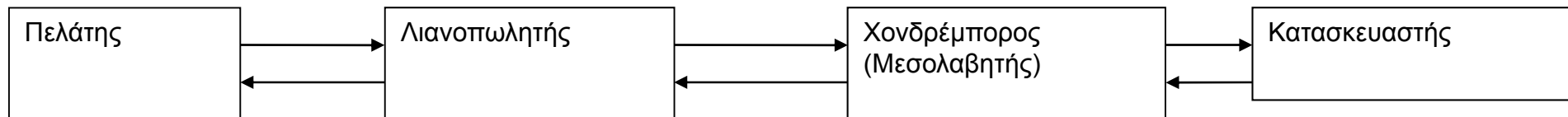
- Εξετάζονται και συγκρίνονται αναλυτικά τα αποθέματα οι ελλειμματικές παραγγελίες (back orders) και το κόστος των μελών της αλυσίδας

## **5. Ανταγωνιστικό μοντέλο**

- Σχεδίαση του ανταγωνιστικού μοντέλου
- Κριτήρια και εύρεση του μέλους-νικητή
- Πολιτικές για διαφοροποίηση του αποτελέσματος

# 1. ΕΦΟΔΙΑΣΤΙΚΗ ΑΛΥΣΙΔΑ

Ο όρος "εφοδιαστική αλυσίδα" (supply chain) περιγράφει το πλέγμα διαδικασιών που απαιτούνται ώστε ένα προϊόν να περάσει από τη φάση της παραγωγής στη φάση της κατανάλωσης.



# Bullwhip effect

## Παρατηρείται σε συστήματα

- Βασισμένα στη πρόβλεψη ζήτησης
- Όπου τα μέλη είναι αλληλοεξαρτώμενα και αλληλοεπιδρώμενα

## Τα αποτελέσματα σε μία SC είναι

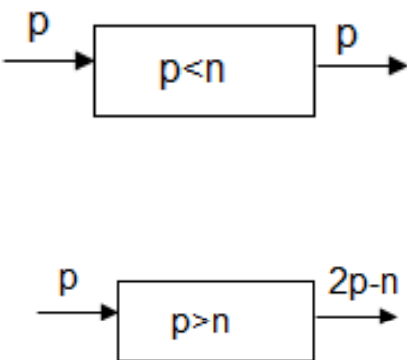
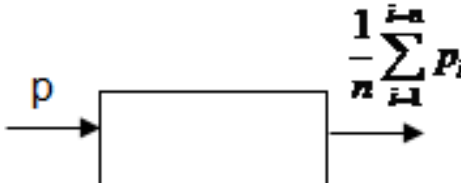
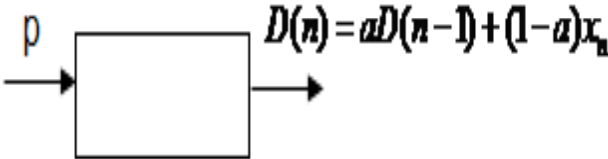
- Μεγάλη αύξηση του τελικού αποθέματος
- Δημιουργία backorders και μηδενικού stock για μεγάλα χρονικά διαστήματα
- Καθυστερήσεις στην παράδοση μιας παραγγελίας
- Αισθητή απόκλιση της πραγματικής ζήτησης από την προβλεπόμενη- αύξηση του κόστους

## Τρόποι αντιμετώπισής του

- Δημιουργία πολλών κέντρων διανομής
- Καταγραφή των παραγγελιών των χαμηλότερων στρωμάτων της αλυσίδας
- Εισαγωγή κατάλληλων **συναρτήσεων πρόβλεψης**



# Εφοδιαστική αλυσίδα με χρήση συναρτήσεων

Aggressive order	Κινητού μέσου όρου (moving average)	Εκθετικής εξομάλυνσης (exponential smoothing)
 <p> <math>p \rightarrow</math> <math>p &lt; n</math> <math>p \rightarrow</math>  <math>p \rightarrow</math> <math>p &gt; n</math> <math>2p - n \rightarrow</math> </p>	<p>με <math>p_i</math> οι παραγγελίες στο <math>i</math>-οστό βήμα</p>  <p> <math>p \rightarrow</math> <math>\frac{1}{n} \sum_{i=1}^{i=n} p_i \rightarrow</math> </p>	<p>με <math>D(n)</math> η ζήτηση στο <math>n</math>-οστό βήμα, α συντελεστής εξομάλυνσης</p> $D(n-1) = \frac{1}{n-1} \sum_{i=1}^{i=n-1} p_i$  <p> <math>p \rightarrow</math> <math>D(n) = aD(n-1) + (1-a)x_n \rightarrow</math> </p>

## 2.PETRI NETS και COLOUR PETRI NETS

- Επινοήθηκαν το 1962 από τον Carl Adam Petri κατά την διδακτορική του διατριβή με θέμα «Επικοινωνία μεταξύ αυτόματων μηχανών»
- Αρχικά χρησιμοποιήθηκε για την μοντελοποίηση και ανάλυση συστημάτων με ταυτόχρονα γεγονότα
  - Στη συνέχεια ανακαλύφθηκε ότι είναι ισχυρό εργαλείο για την περιγραφή και τον έλεγχο συστημάτων που εξαρτώνται από γεγονότα, όπως ασύγχρονα, παράλληλα, με συγκρούσεις αμοιβαίου αποκλεισμού

Ένα Petri δίκτυο μπορεί θεωρηθεί μια πεντάδα  $M=(P,T,I,O,\mu)$

$P$ : το σύνολο των καταστάσεων ( $p$ )(places)

$T$ : Το σύνολο των μεταβάσεων ( $t$ )(transition)

$I$ : Συνάρτηση εισόδου (input function) με  $I: T \rightarrow P$

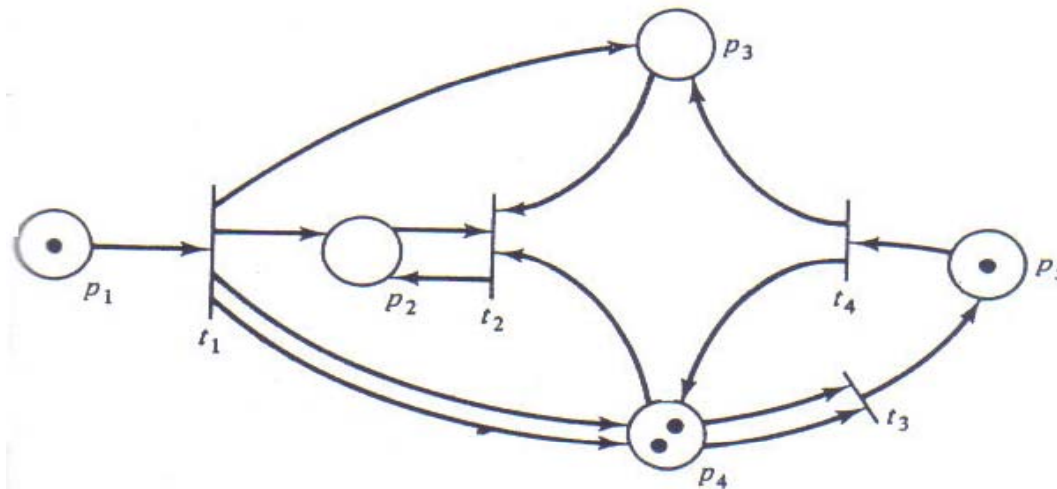
$O$ : Συνάρτηση εξόδου (output function) με  $O: T \rightarrow P$

όπου

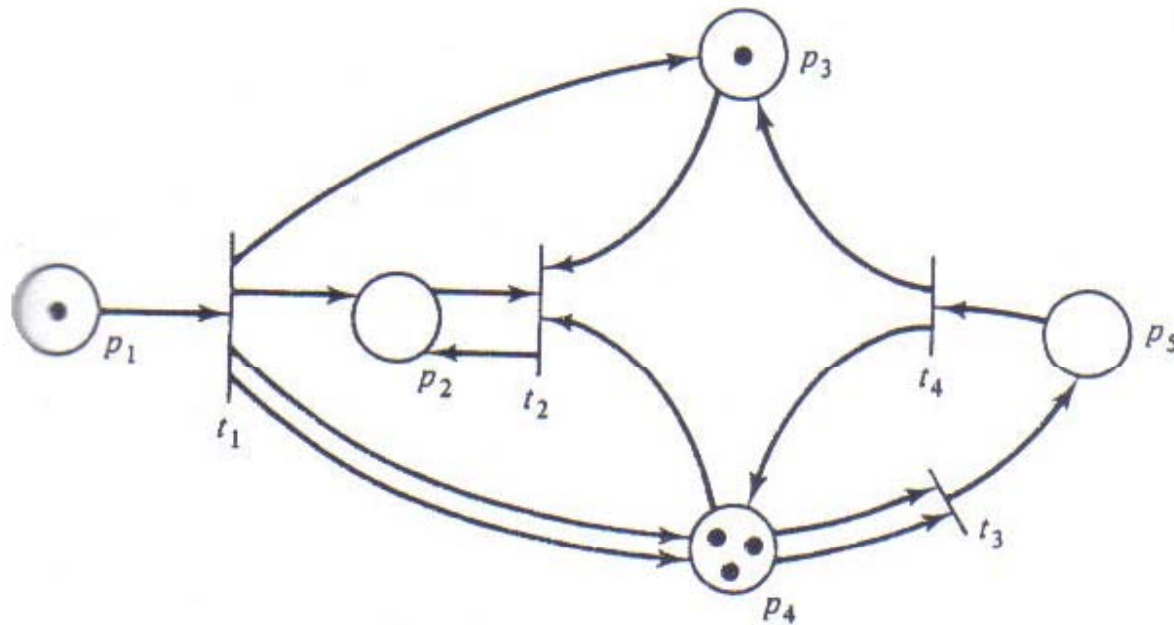
στην  $I(t_i)=p_i$  οι  $p_i$  είναι οι καταστάσεις που ξεκινούν από την μετάβαση  $t_i$ .

στην  $O(t_j)=p_j$  οι  $p_j$  είναι οι καταστάσεις που θα καταλήξει η  $t_j$ .

$\mu$ : συνάρτηση που υπολογίζει το πλήθος των μαρκών (tokens) της κάθε κατάστασης  $P \rightarrow N$



# Ιδιότητες των PN

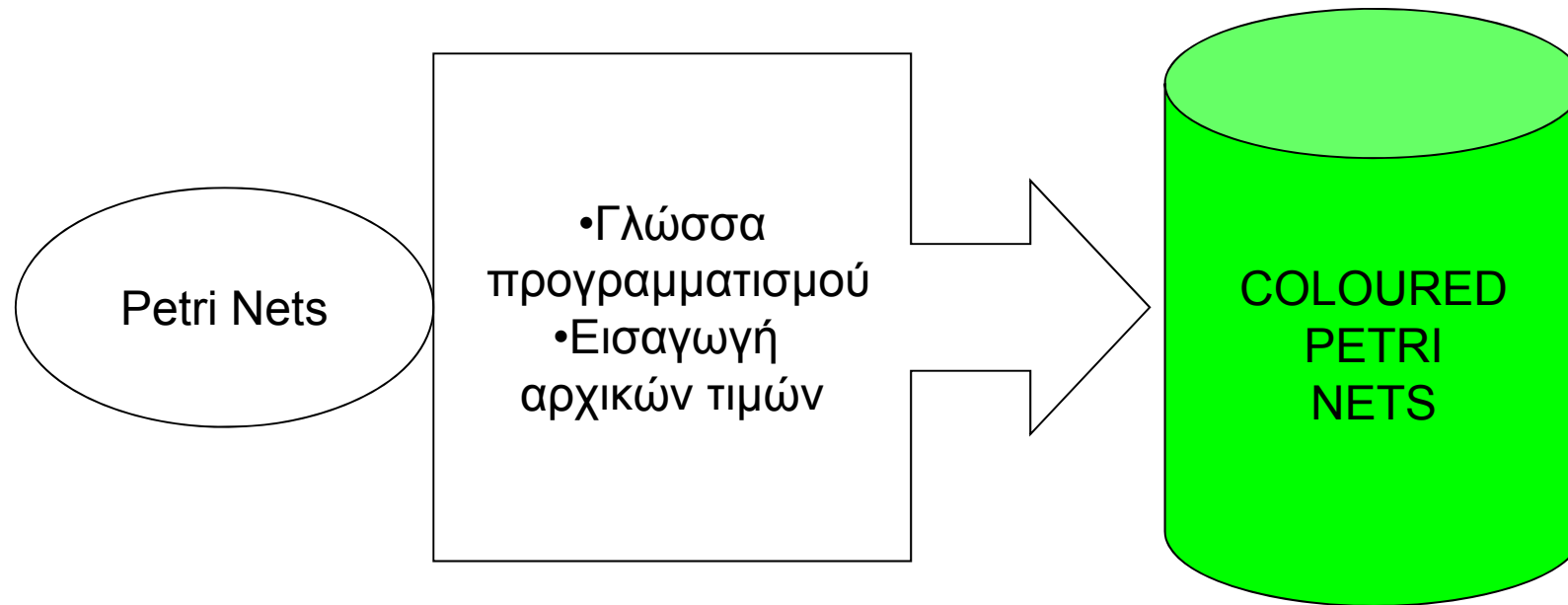


- Μαρκάρισμα
- Εκτελεσιμότητα
- Περιοριστικότητα

## Μειονεκτήματα των απλών PN

- Δεν υπάρχει η δυνατότητα επεξεργασίας δεδομένων
- Αδυναμία σχεδιασμού **ιεραρχικών δικτύων.**

# Μετάβαση από τα PN στα CPN



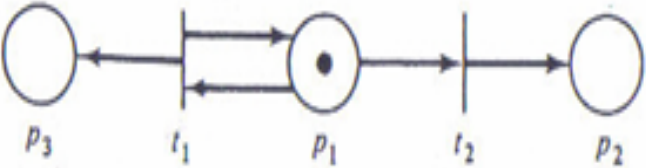
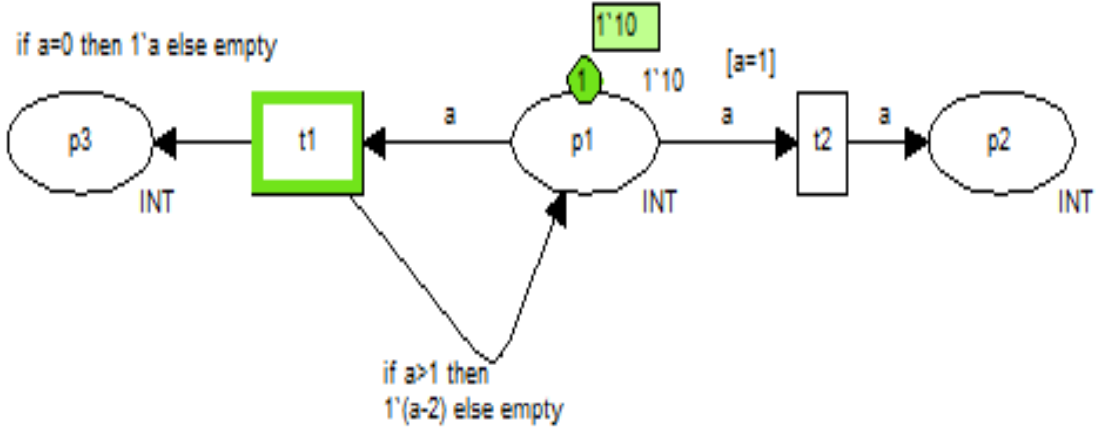
# CPN

Με χρήση των CPN μπορούμε να ελέγξουμε

- το επίπεδο αποθεμάτων,
- τις εισερχόμενες παραγγελίες
- τη συμπεριφορά του συστήματος

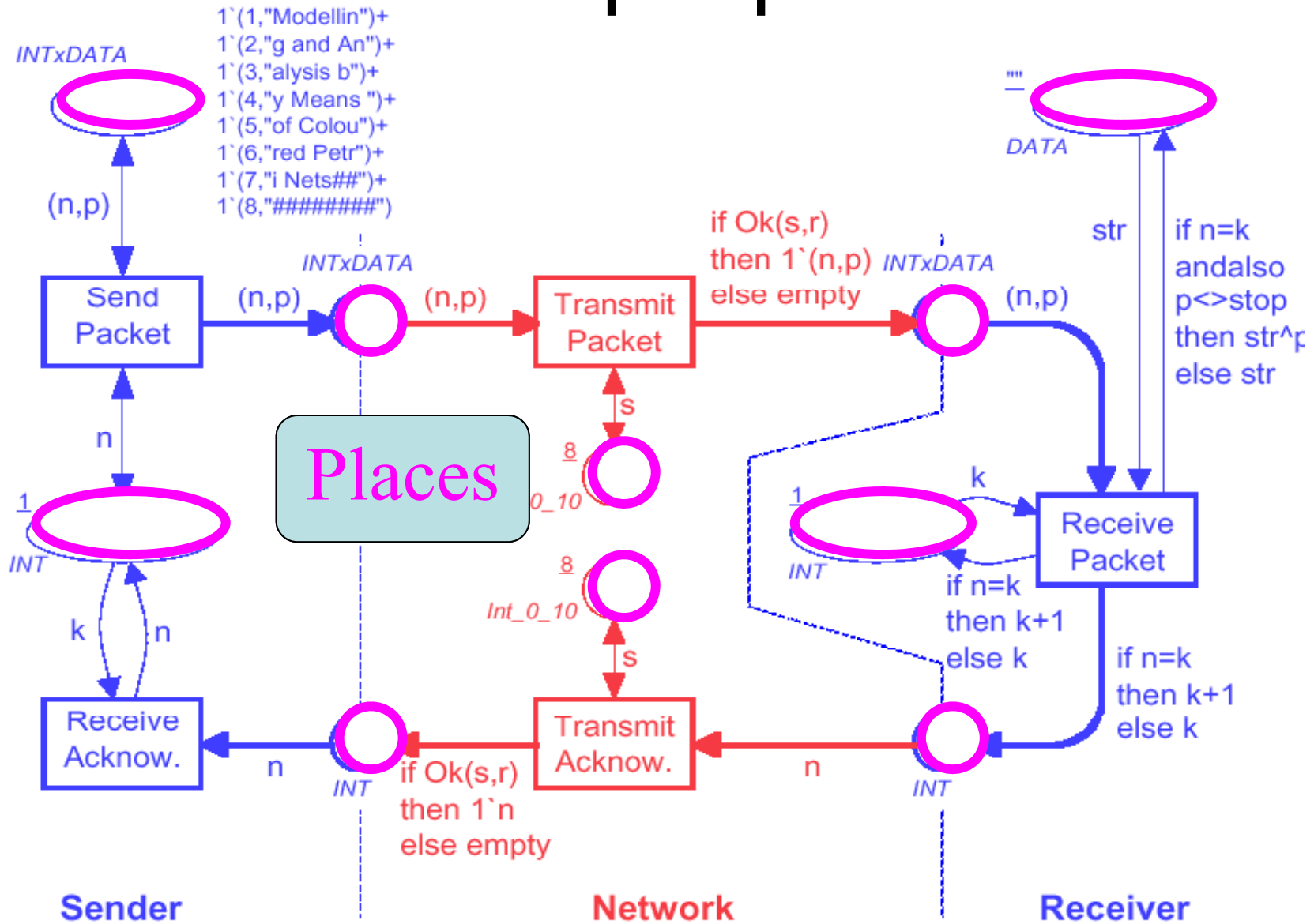
Μπορούμε εύκολα να επέμβουμε κατά τη διάρκεια της προσομοίωσης.

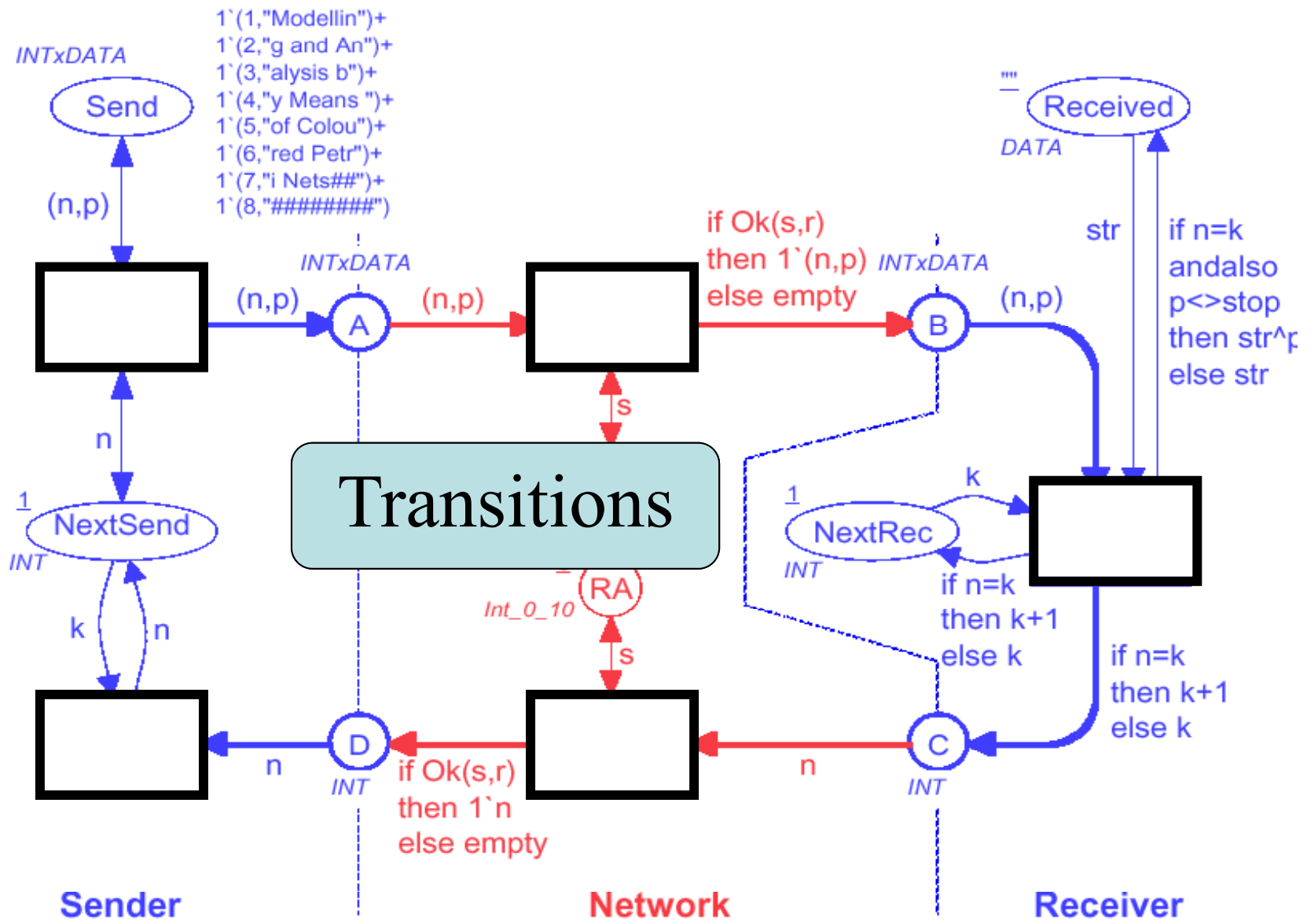
# ΒΑΣΙΚΕΣ ΔΙΑΦΟΡΕΣ ΤΩΝ PN και CPN

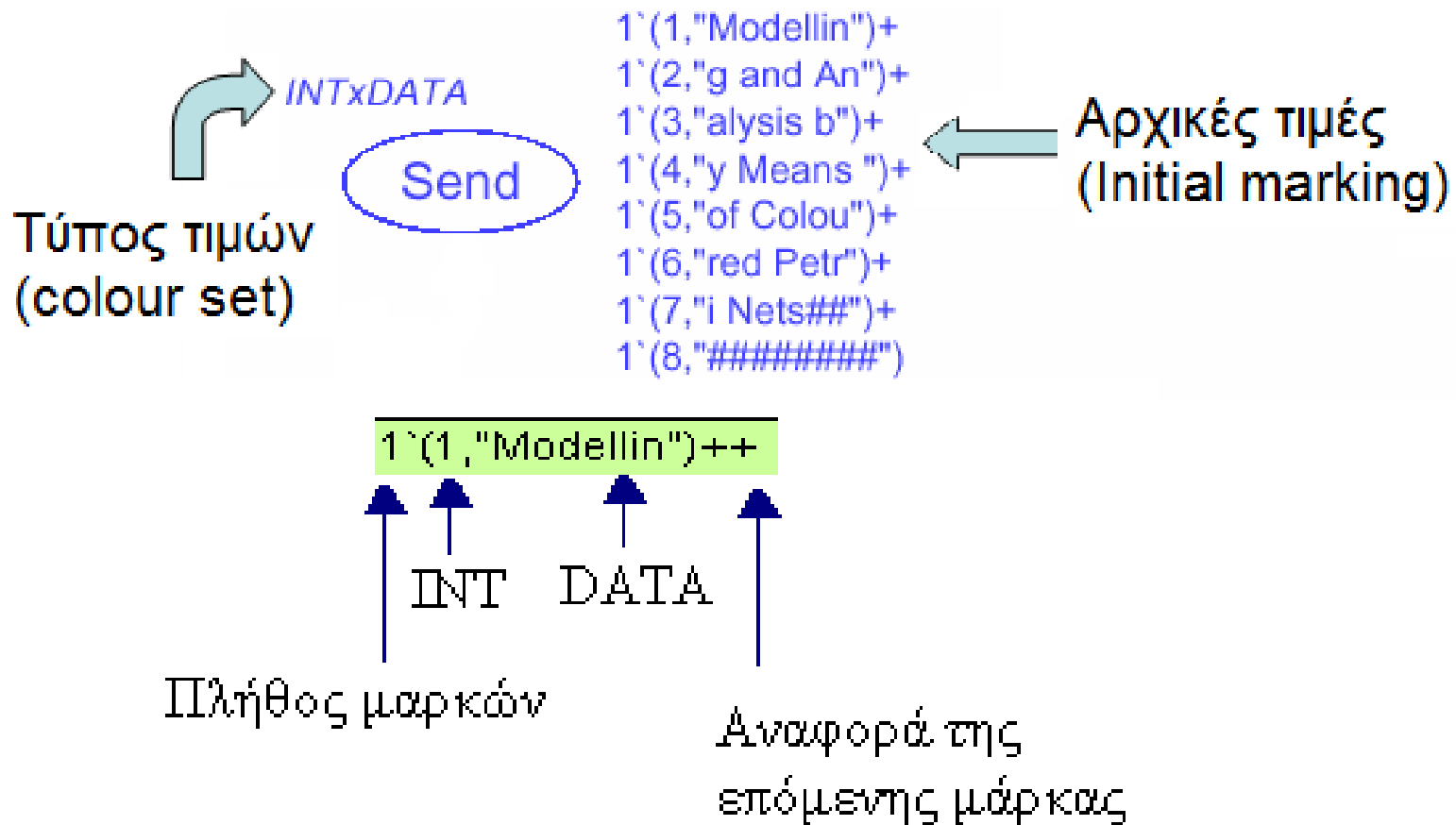
Petri Nets	Coloured Petri Nets
	<p>if a=0 then 1'a else empty</p>  <p>if a&gt;1 then 1'(a-2) else empty</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p>▼ Declarations</p> <p>▼ Standard declarations</p> <p>▼ colset INT = int;</p> <p>▼ var a:INT;</p> </div>
<p>Αρχική τιμή με μορφή μάρκας στην p1</p>	<p>Αρχική τιμή με μορφή μεταβλητής τύπου INT, BOOL, STRING, DATA κτλ</p>
<p>Η πυροδότηση στην επόμενη ενεργή μετάβαση είναι τυχαία</p>	<p>Η πυροδότηση γίνεται με βάση εντολών</p>
	<p>Χρήση γλώσσας προγραμματισμού</p>



# Simple protocol







# Δίκτυο αποστολής πακέτων

Το δίκτυο που ακολουθεί περιγράφει τη μεταφορά πακέτων από τον αποστολέα στον παραλήπτη όπου μπορεί ανάλογα με τη διαδρομή που θα χρησιμοποιηθεί το πακέτο να παραδοθεί στην ώρα του, να αργήσει ή και να χαθεί.

▶ Tool box

▶ Help

▶ Options

▼ SimpleProtocol.cpn

Step: 0

Time: 0

▶ Options

▶ History

▼ Declarations

▼ colset INT = int;

▼ colset DATA = string;

▼ colset INTxDATA = product INT \* DATA;

▼ var n, k: INT;

▼ var p, str: DATA;

▼ val stop = "#####";

▼ colset Ten0 = int with 0..10;

▼ colset Ten1 = int with 1..10;

▼ var s: Ten0;

▼ var r: Ten1;

▼ fun Ok(s:Ten0,r:Ten1) = (r<=s);

▼ Monitors

Top

Binder 0

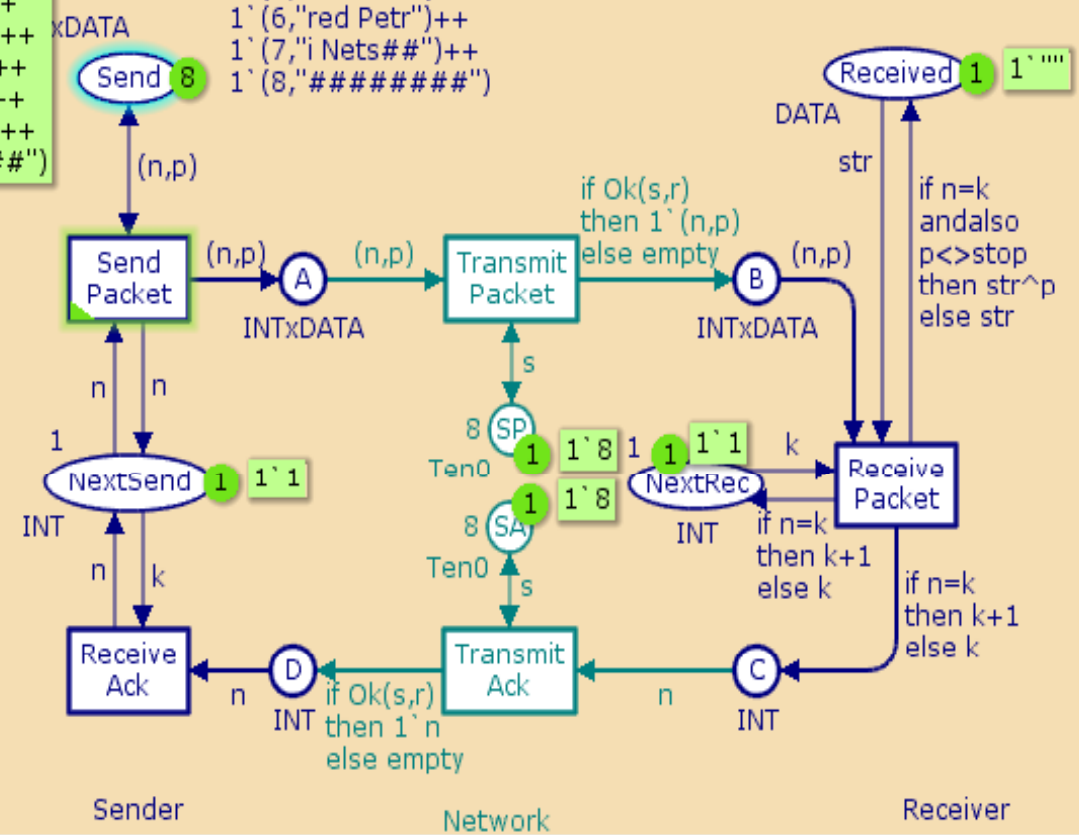
Top

```

1` (1,"Modellin")++
1` (2,"g and An")++
1` (3,"alysis b")++
1` (4,"y Means ")++
1` (5,"of Colou")++
1` (6,"red Petr")++
1` (7,"i Nets##")++
1` (8,"#####")
    
```

```

1` (1,"Modellin")++
1` (2,"g and An")++
1` (3,"alysis b")++
1` (4,"y Means ")++
1` (5,"of Colou")++
1` (6,"red Petr")++
1` (7,"i Nets##")++
1` (8,"#####")
    
```



None

# Χειροκίνητη προσομοίωση

## Παλέτα χειροκίνητης προσομοίωσης

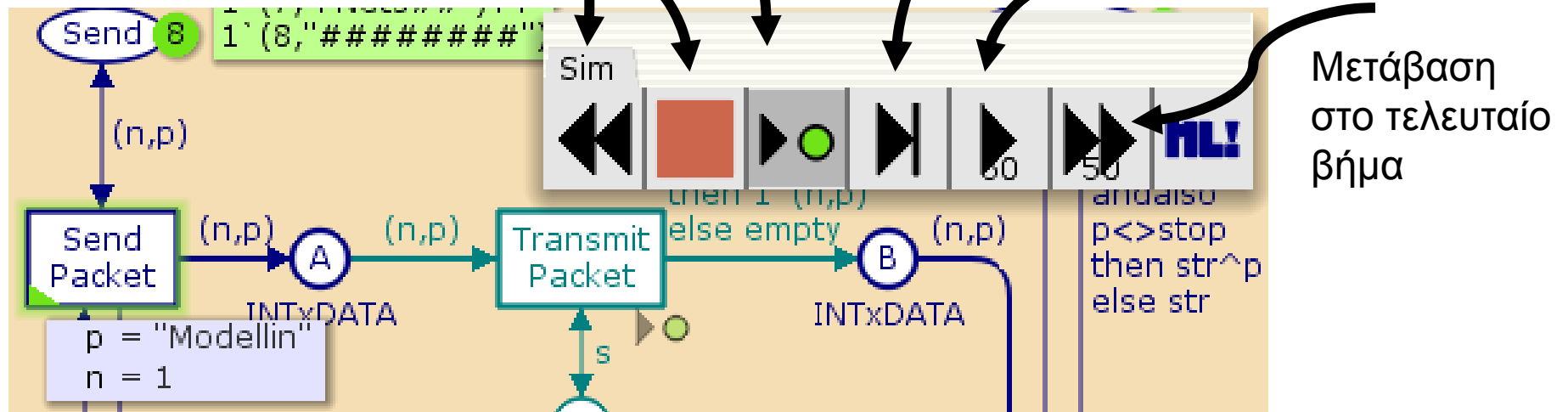
Επιλογή του επόμενου μαρκάριατος και προβολή των τιμών των μεταβλητών

Μετάβαση στο επόμενο βήμα επιλέγοντας το μαρκάρισμα αυτόματα

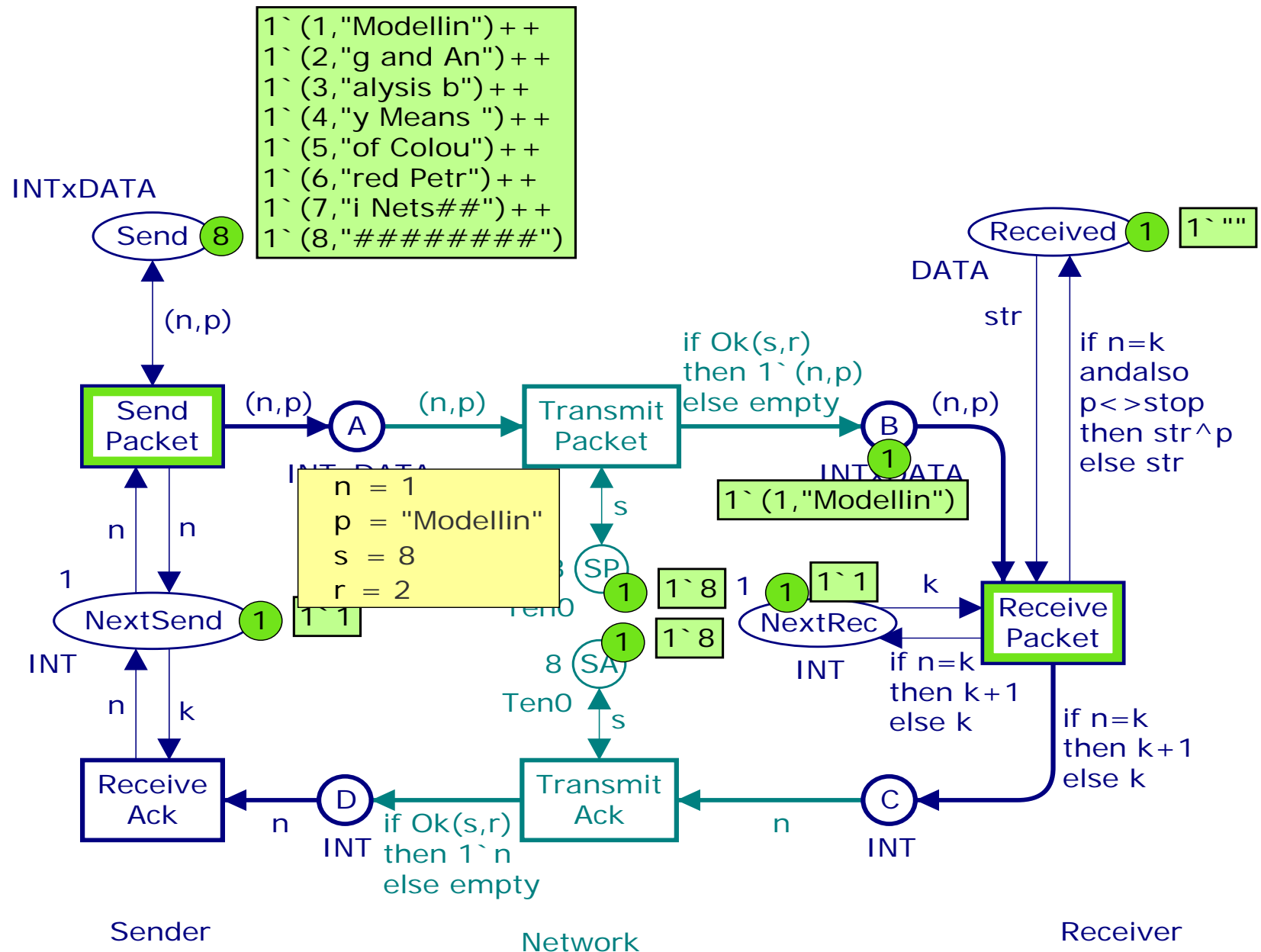
Επανεκκίνηση της προσομοίωσης

παύση

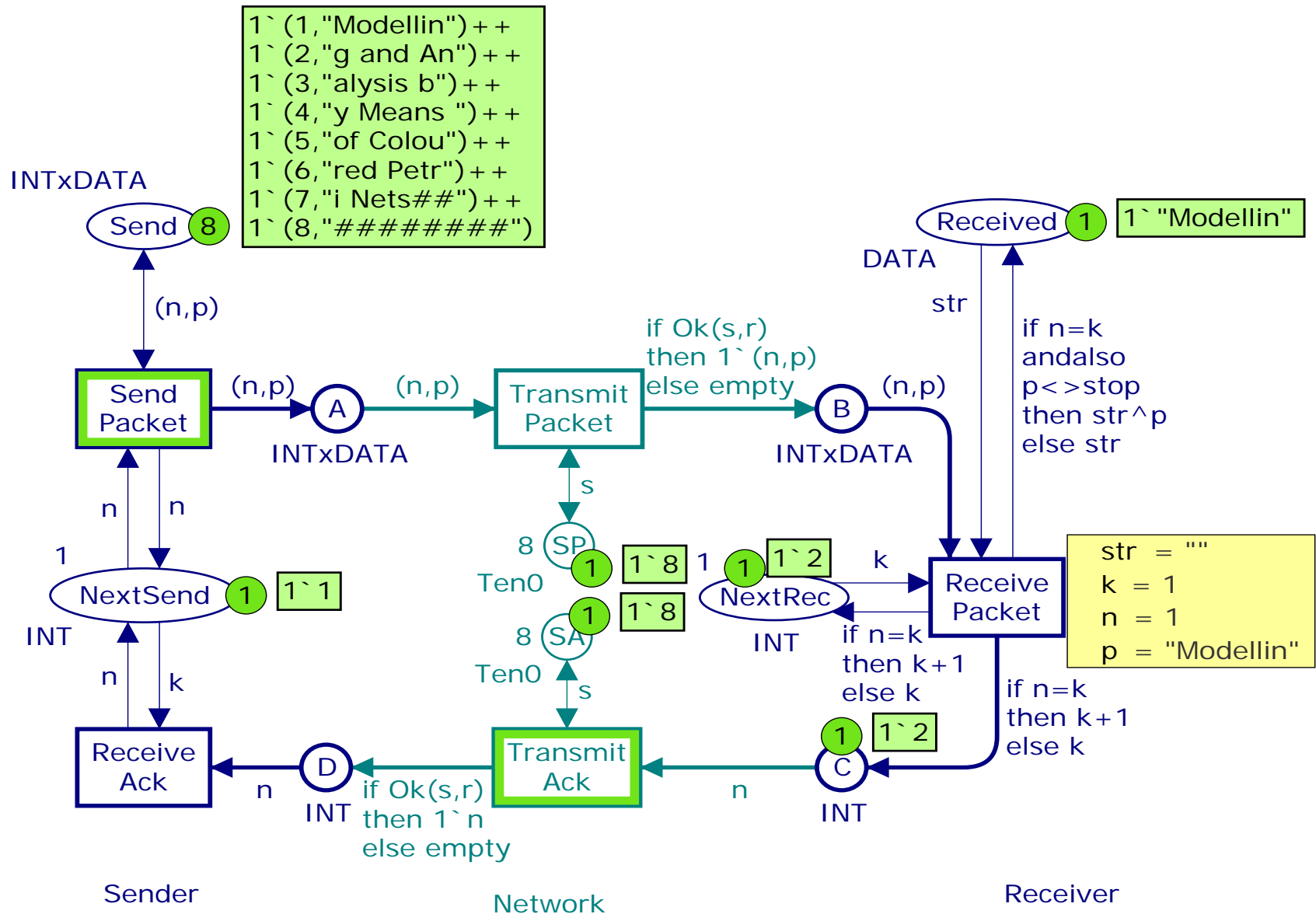
Η προσομοίωση γίνεται αυτόματα μέχρι το τέλος



Μετάβαση στο τελευταίο βήμα



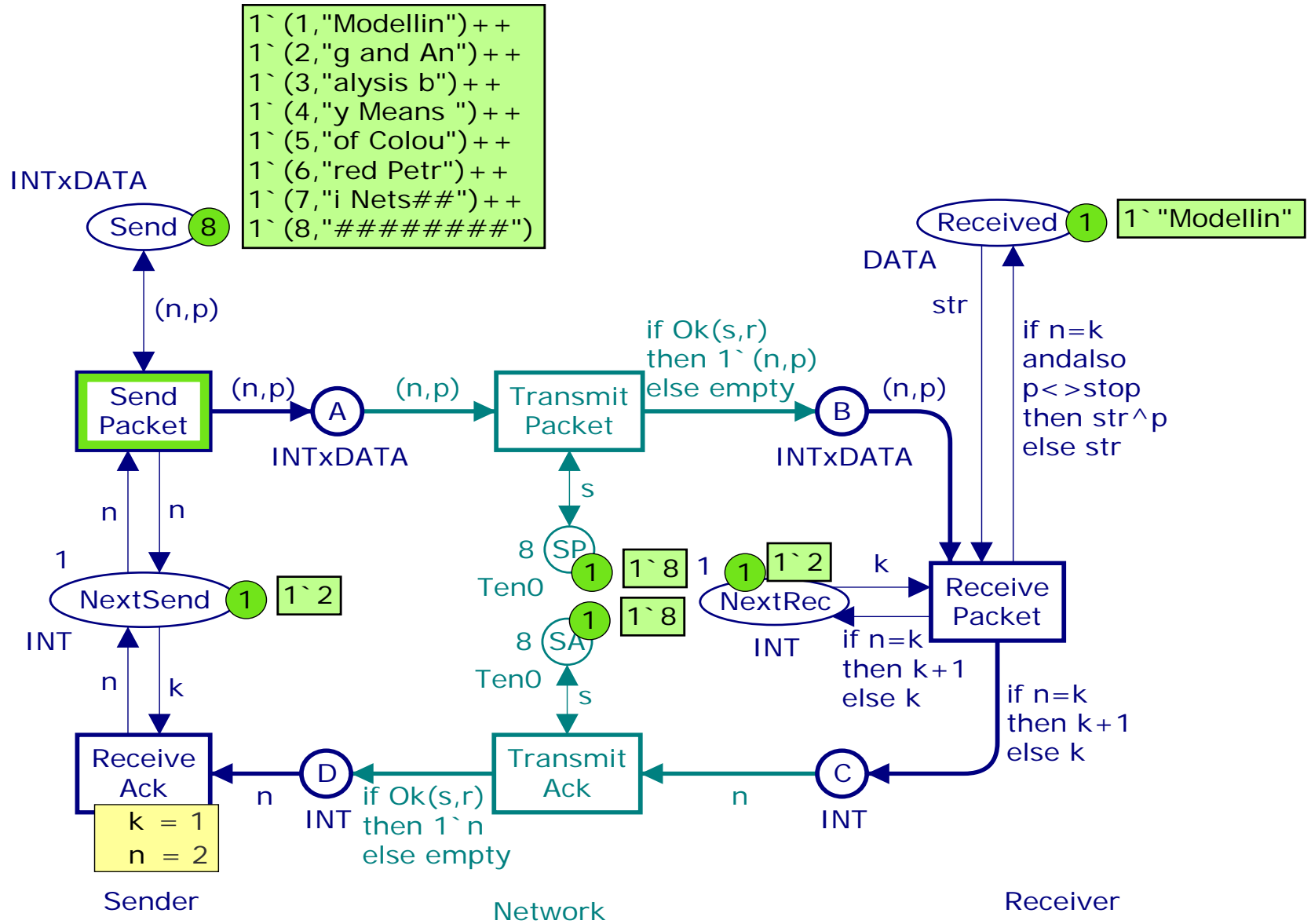
Simple Protocol



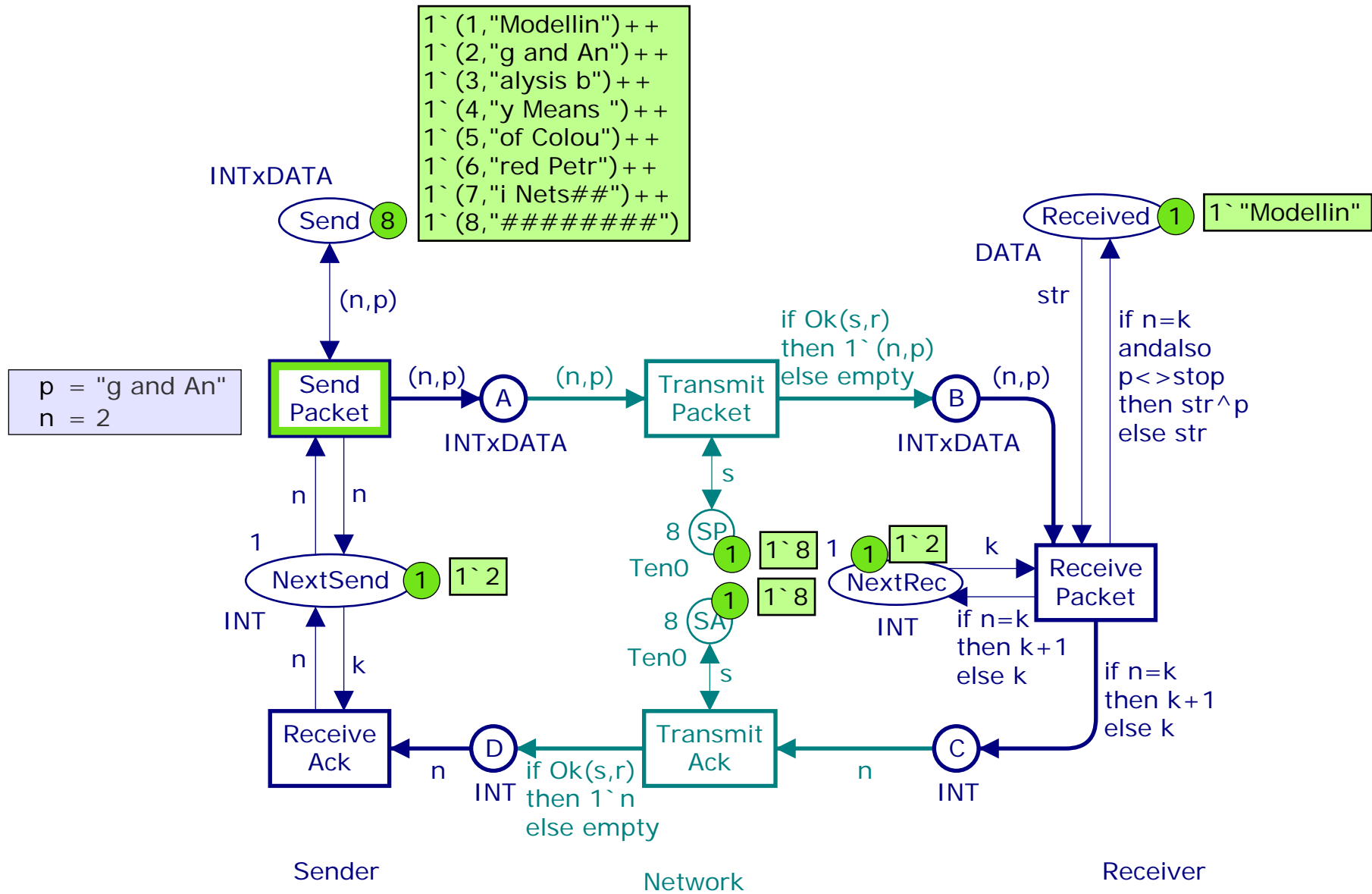
Simple Protocol



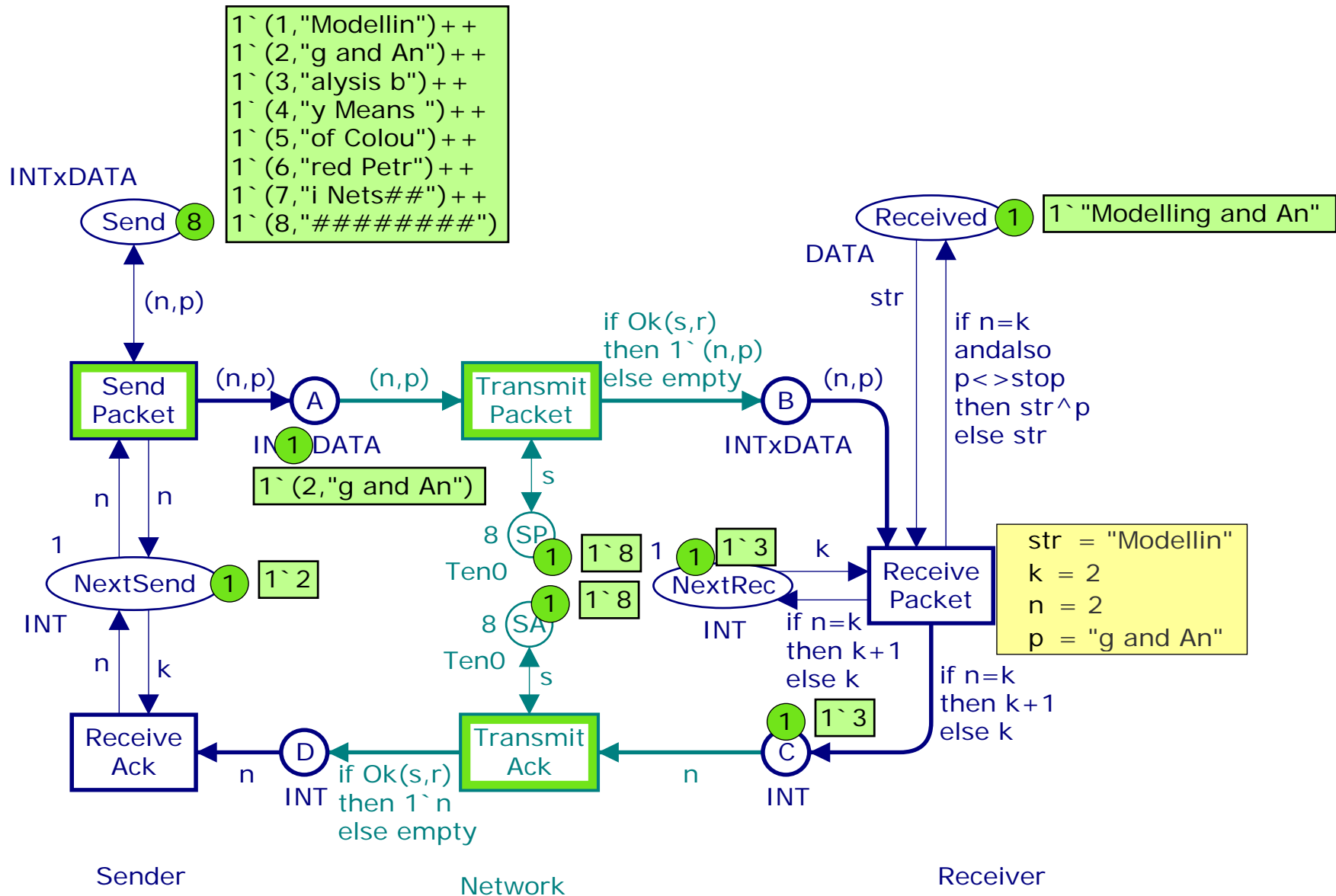




Simple Protocol

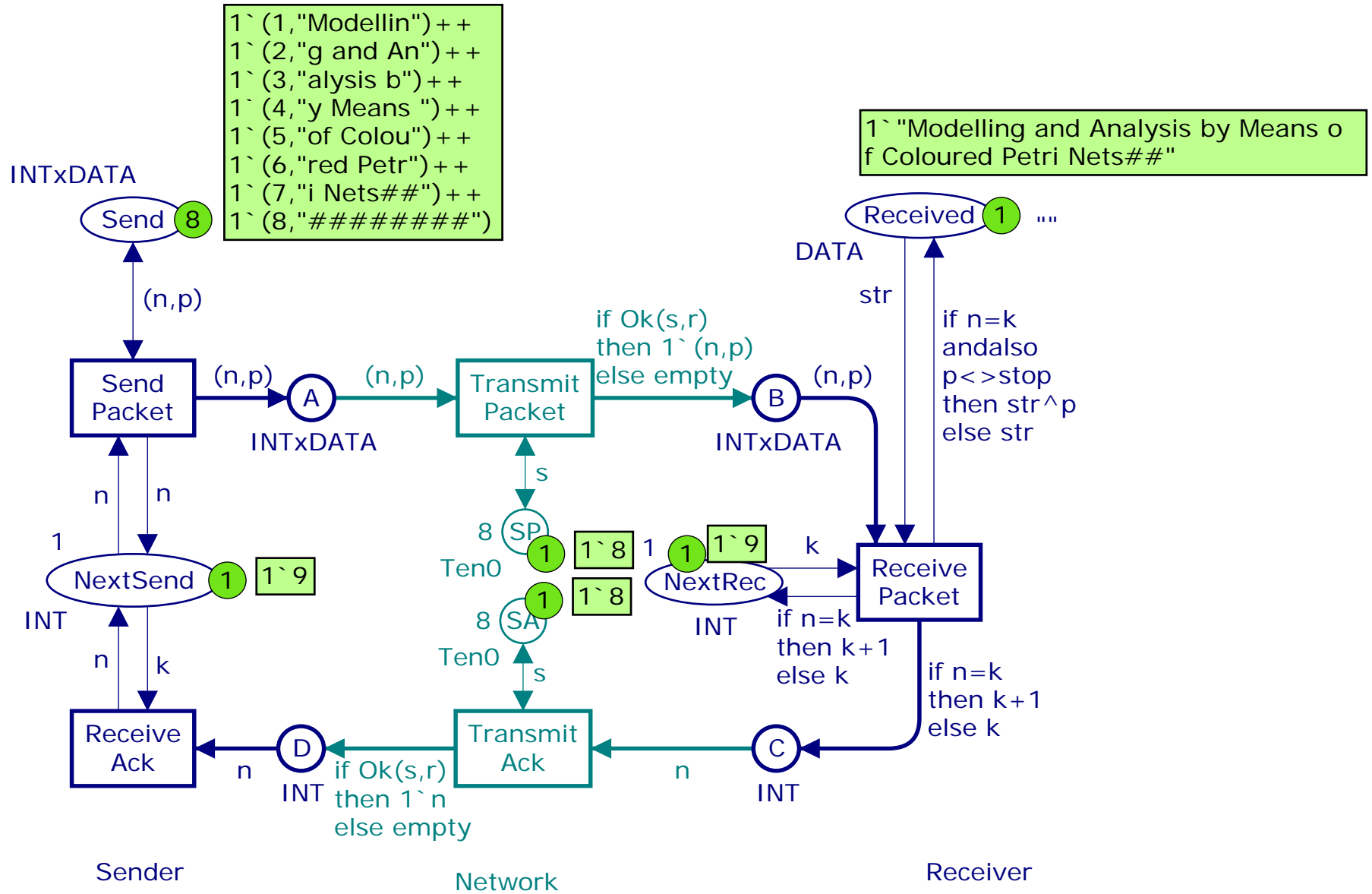


Simple Protocol



Simple Protocol





Simple Protocol

## ΑΝΑΛΥΣΗ ΠΡΟΣΟΜΕΙΩΣΗΣ (simulation report)

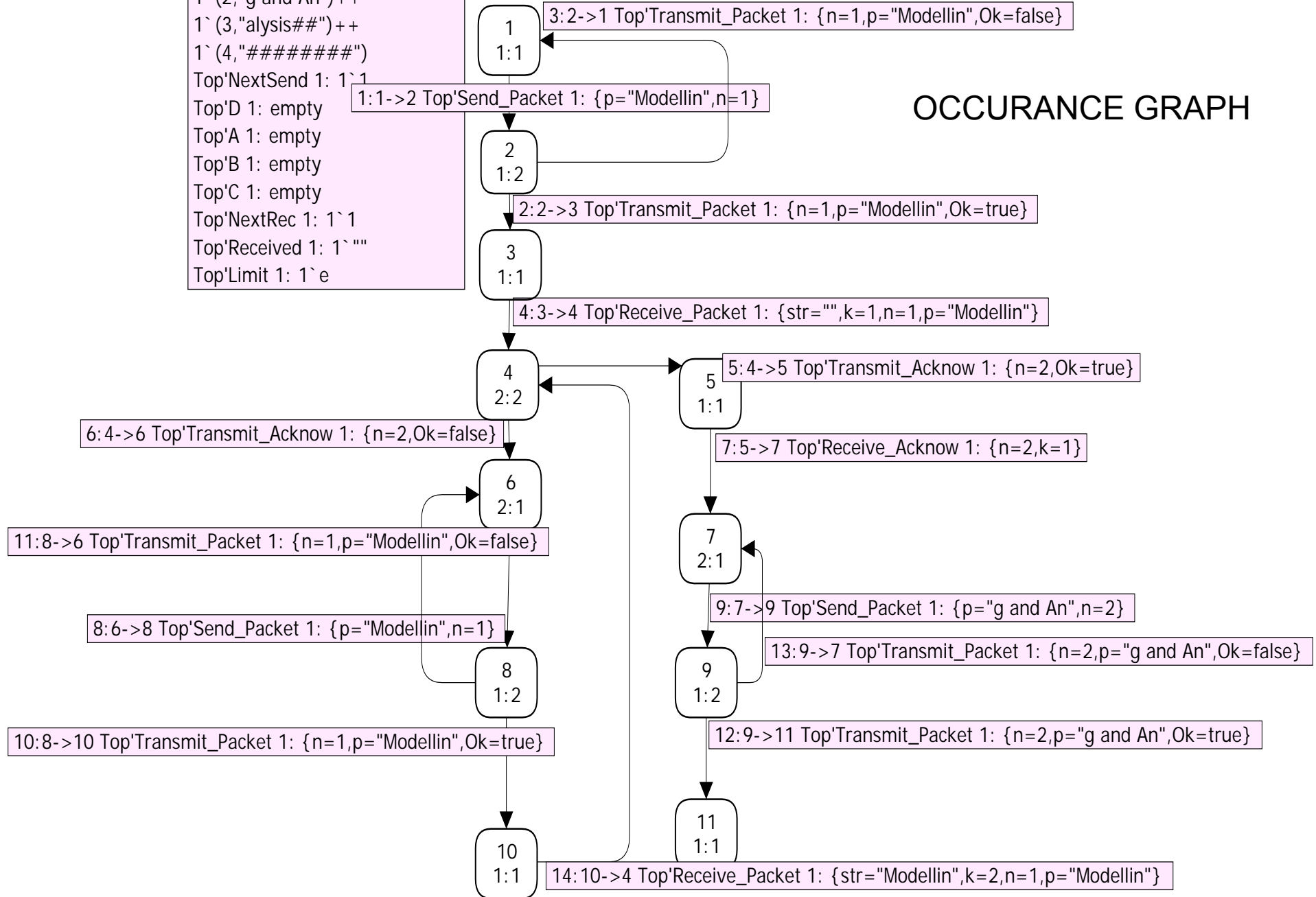
- 1 0 Send\_Packet @ (1:Top)
- - p = "Modellin"
- - n = 1
- 2 0 Transmit\_Packet @ (1:Top)
- - n = 1
- - p = "Modellin"
- - s = 8
- - r = 7
- 3 0 Send\_Packet @ (1:Top)
- - p = "Modellin"
- - n = 1

```

1:
Top'Send 1: 1` (1,"Modellin")+
1` (2,"g and An")+
1` (3,"alysis##")+
1` (4,"#####")
Top'NextSend 1: 1`1
Top'D 1: empty
Top'A 1: empty
Top'B 1: empty
Top'C 1: empty
Top'NextRec 1: 1`1
Top'Received 1: 1`""
Top'Limit 1: 1`e

```

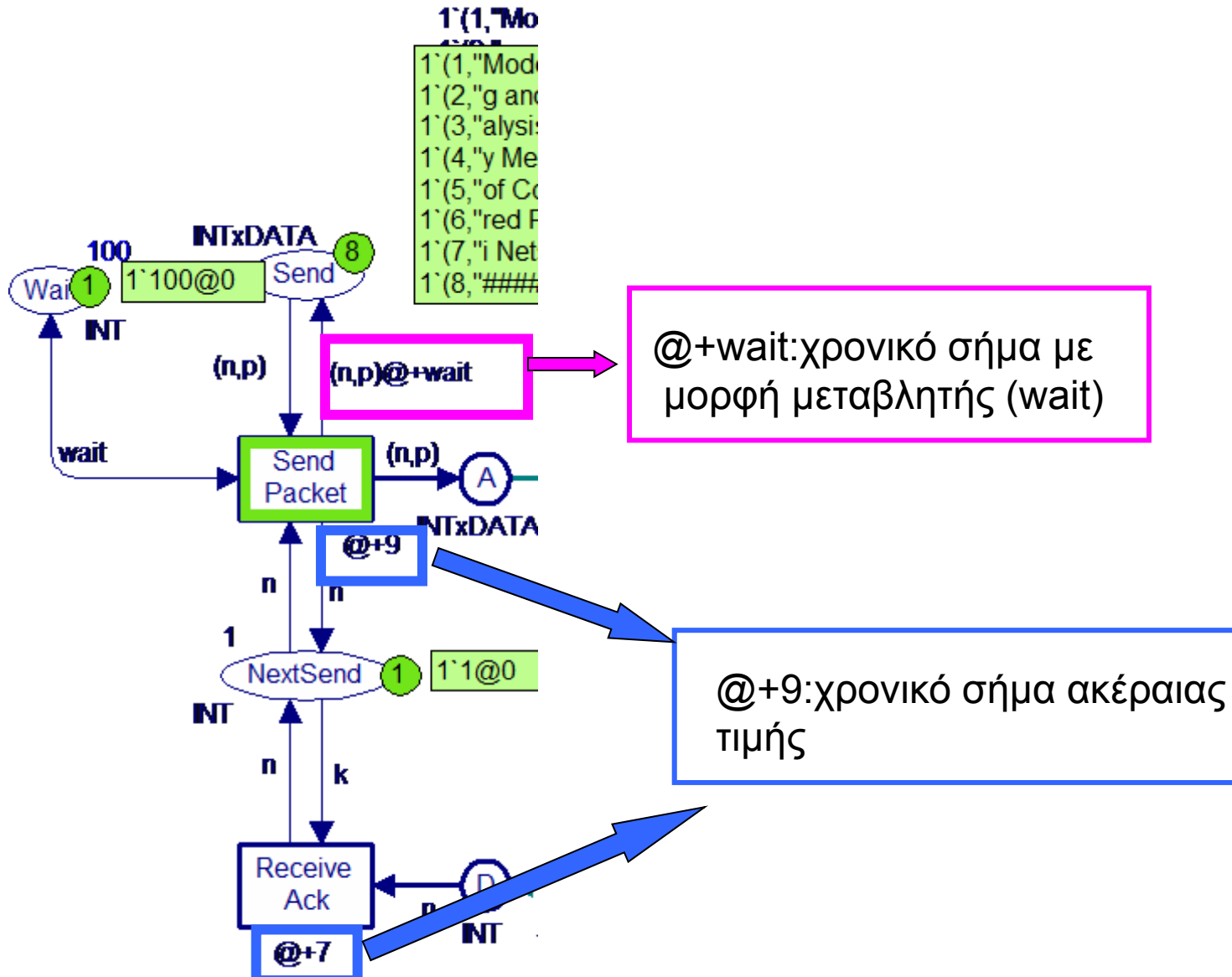
# OCCURANCE GRAPH



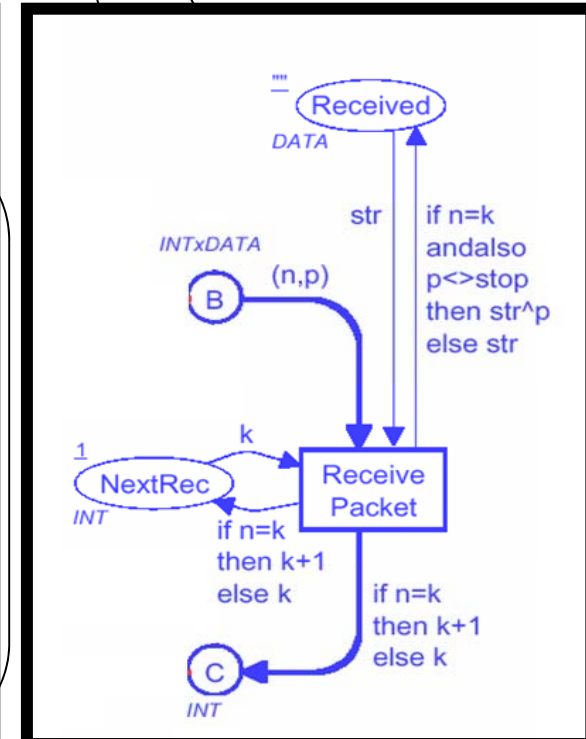
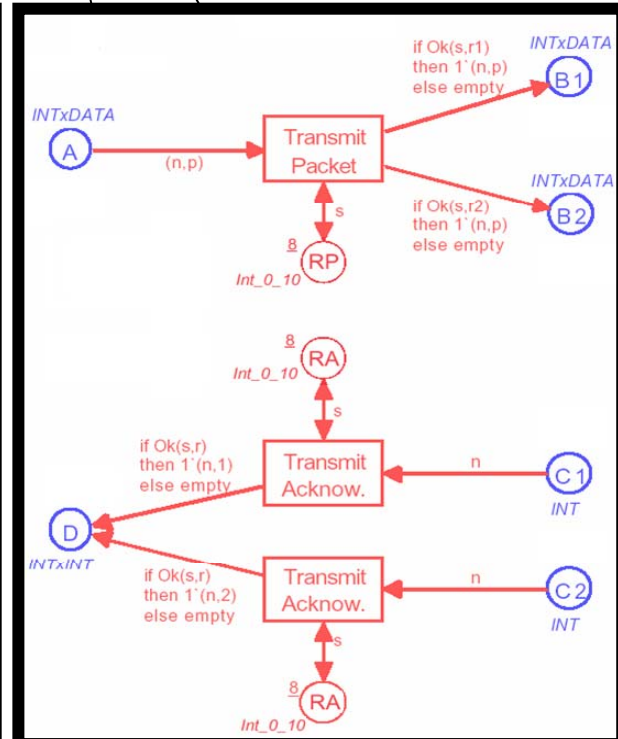
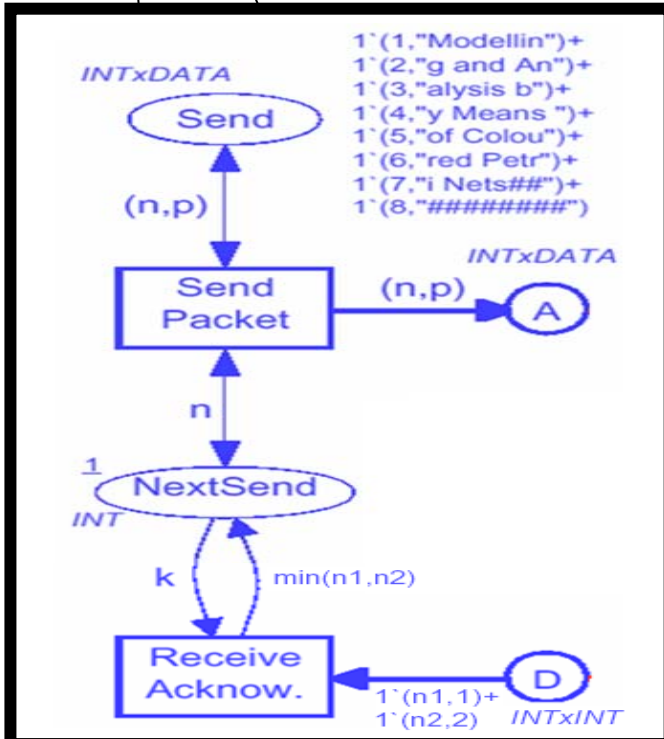
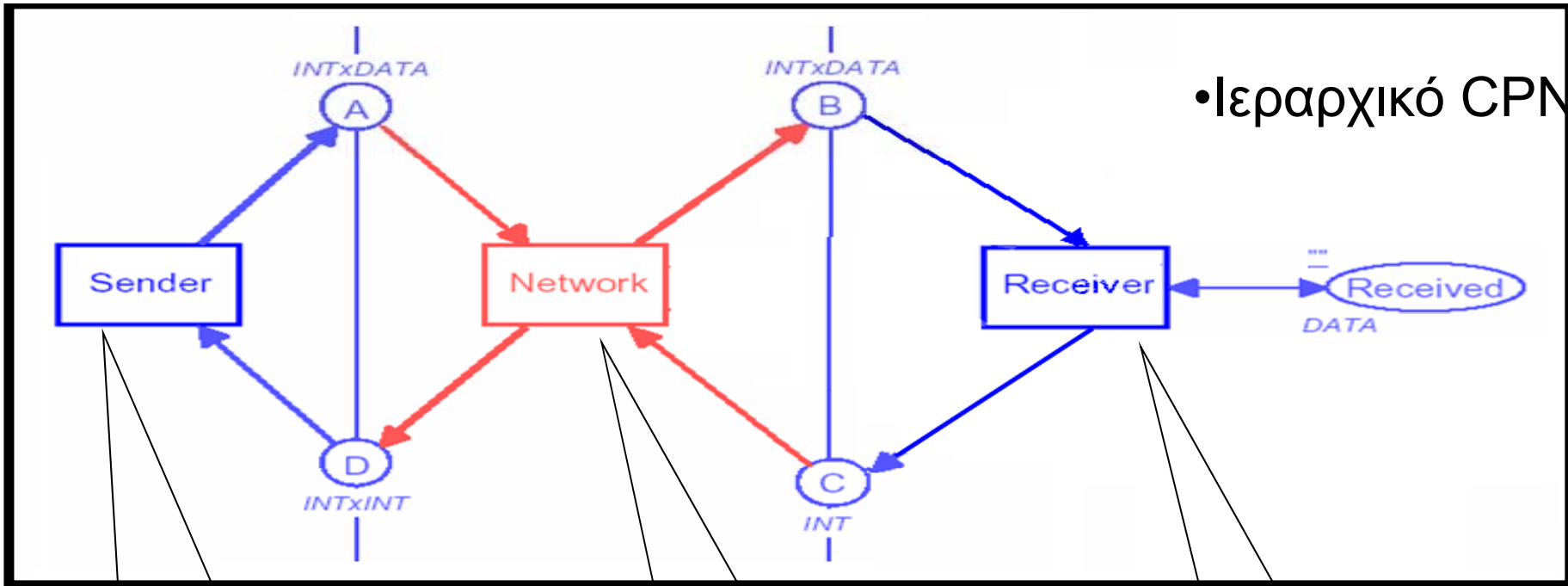


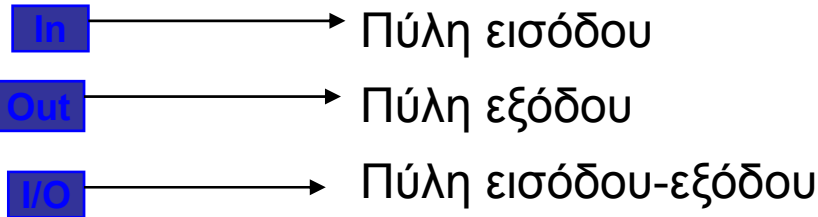
# Είδη CPN

- TIMED CPN

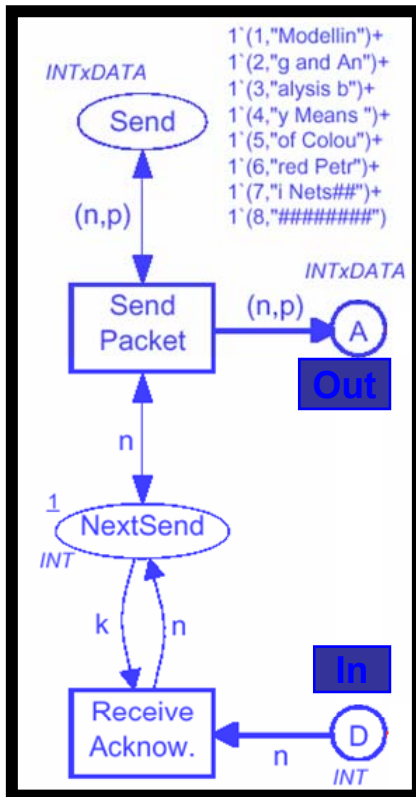


• Ιεραρχικό CPN

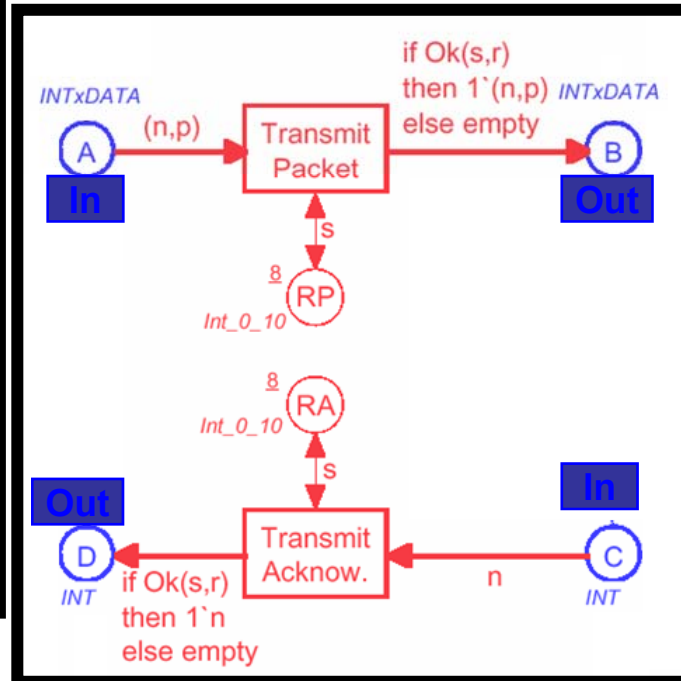




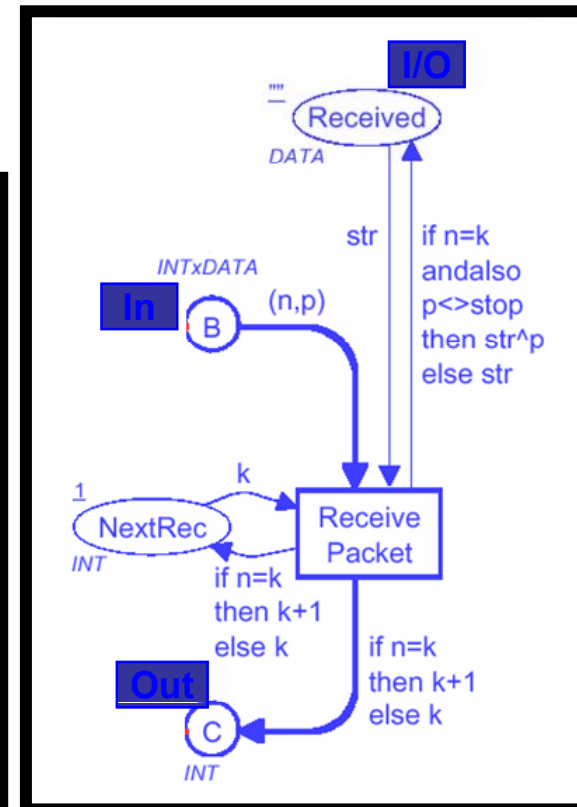
## Sender



## Network



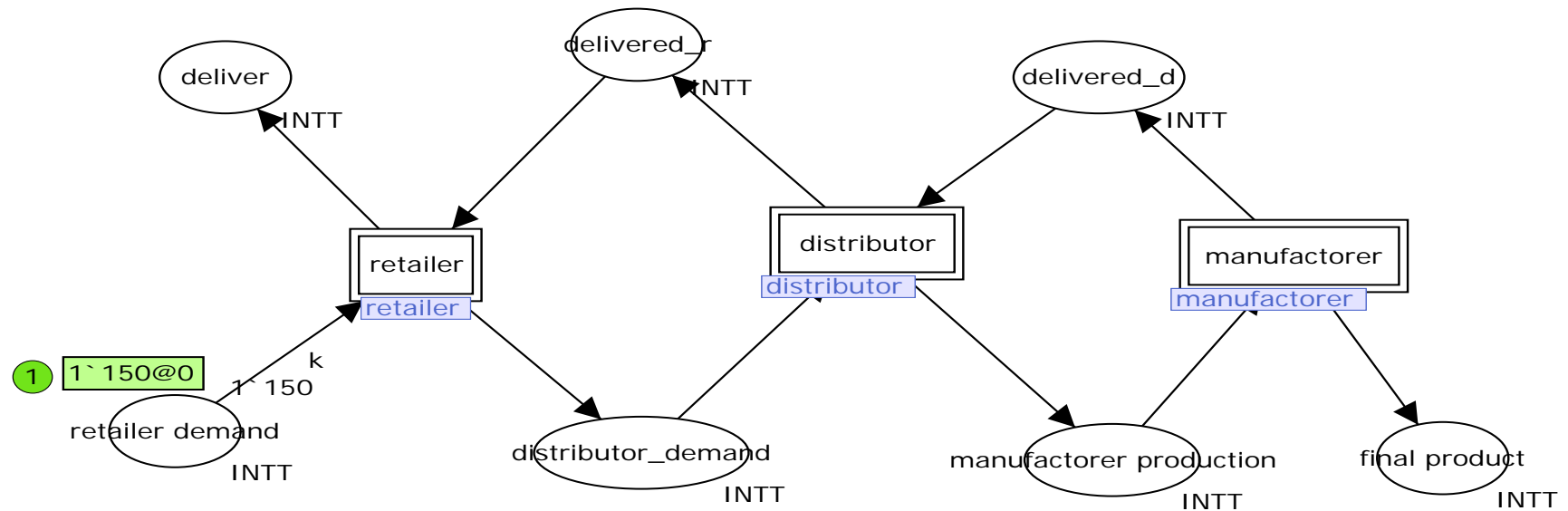
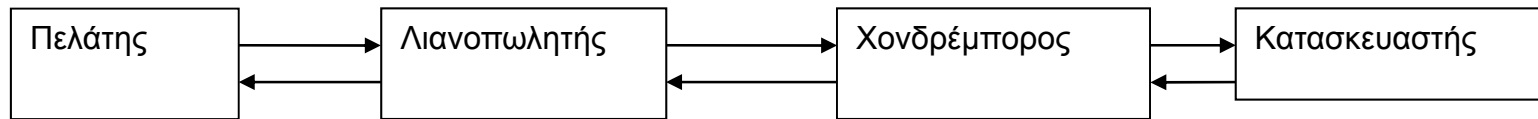
## Receiver



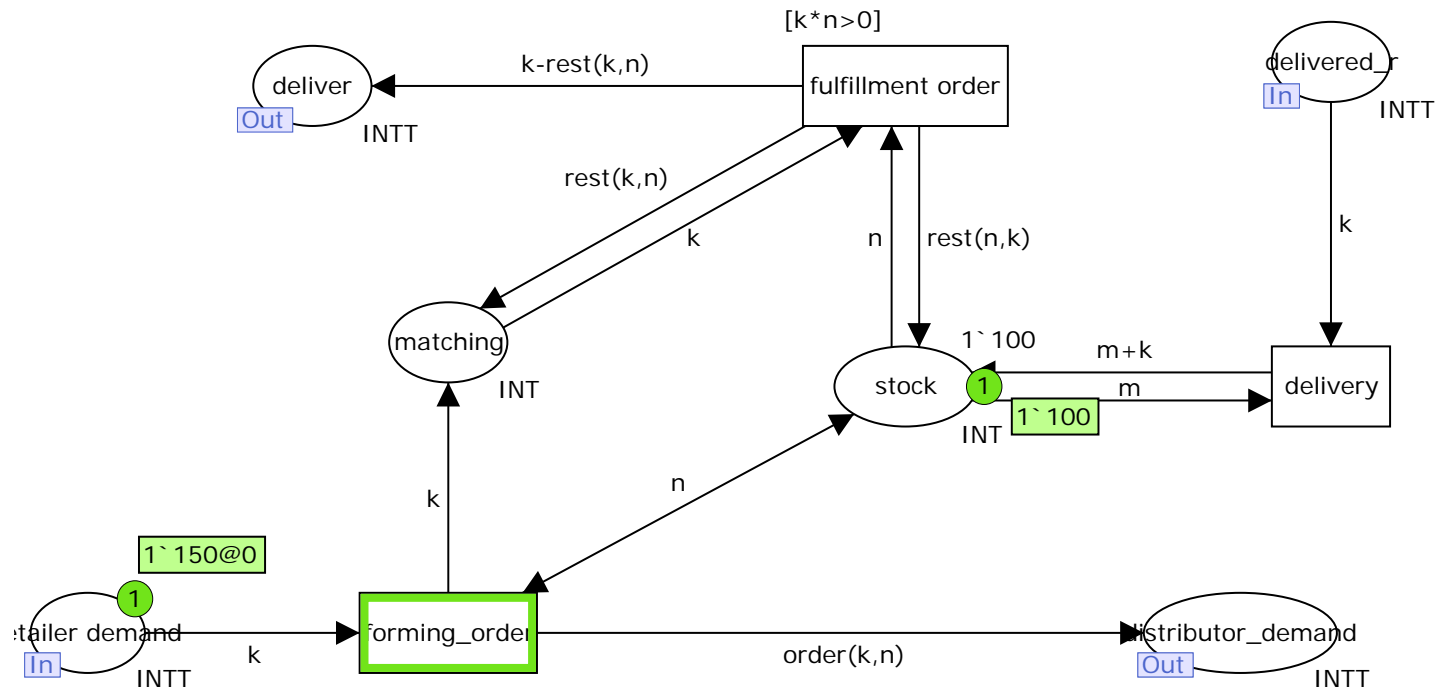
## Πλεονεκτήματα χρήσης ιεραρχικών μοντέλων

- Ευκολία στον χρήστη ως προς τον σχεδιασμό και για τον αναγνώστη ως προς την κατανόηση
- Μείωση πολυπλοκότητας του κεντρικού δικτύου και των υποσελίδων σε σχέση με το αρχικό-μείωση της έκτασης εντολών
- Εύκολη προσαρμογή σε αλλαγή σεναρίου
- Δυνατότητα μετατροπής τους σε ένα απλό CPN

# 3.Εφοδιαστική αλυσίδα σχεδιασμένη από CPN



# Υποσελίδα του “λιανέμπορου”



“retailer demand”

“distributor demand”

“stock”

“matching”

“delivered\_r”

“deliver”

Οι παραγγελίες του πελάτη

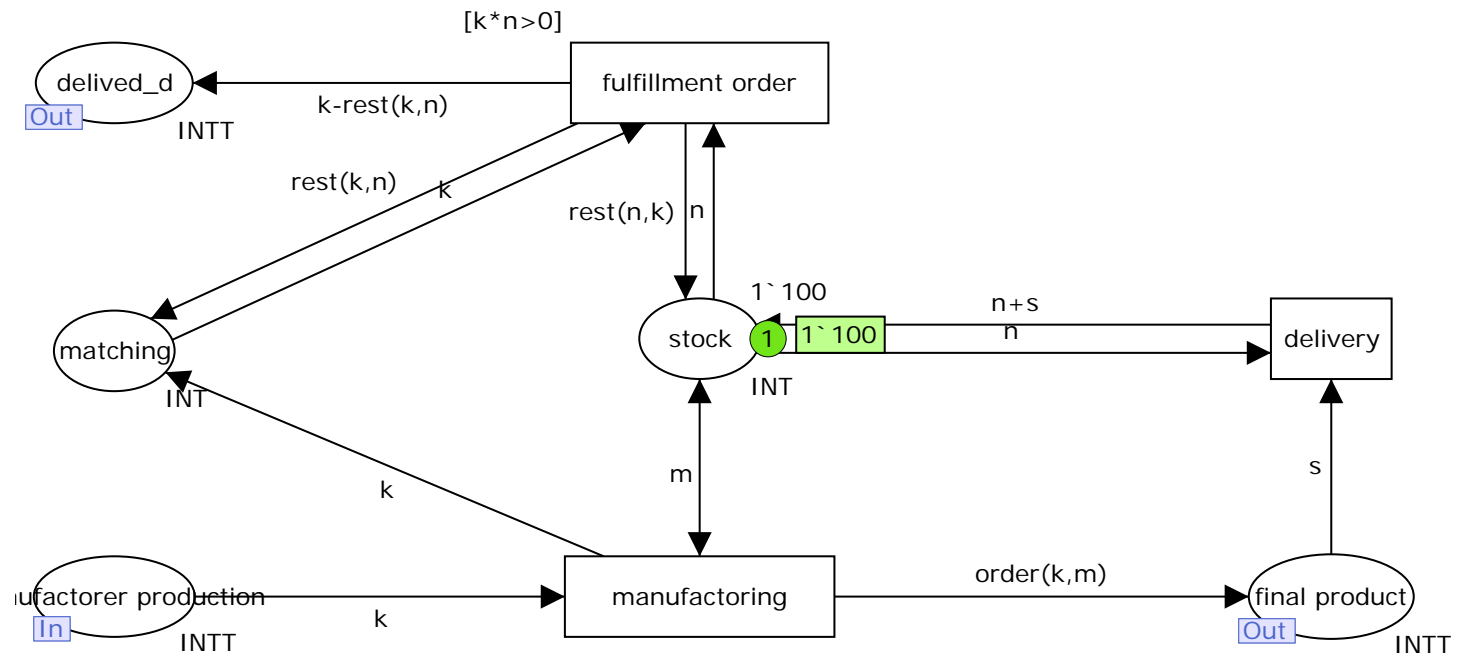
Η εξερχόμενη παραγγελία του λιανέμπορου

Το απόθεμα

Backorders

Οι παραγγελίες που παραλαμβάνει από τον μεσολαβητή

Τα προϊόντα που στέλνει στον πελάτη

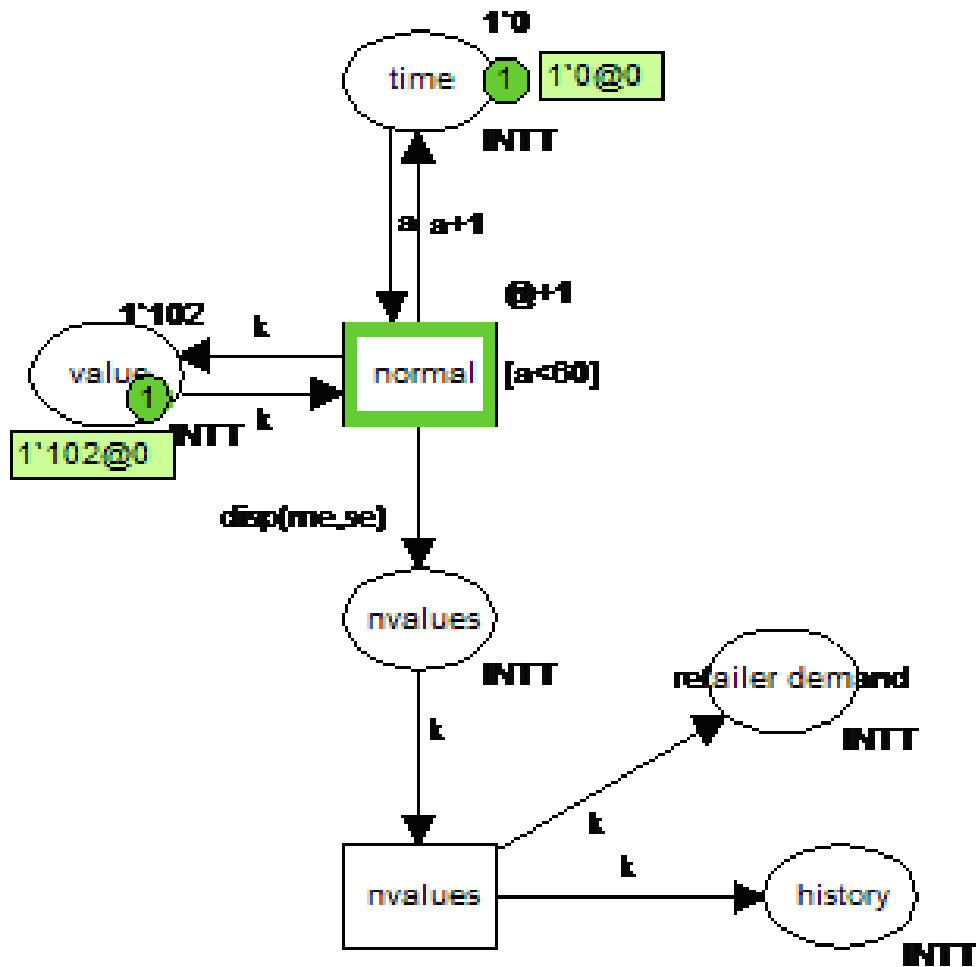


Υποσελίδα του  
“κατασκευαστή”

```

Declarations
▼ val se=33.0;
▼ val me=100.0;
▼ colset INT=int;
▼ colset INTT=INT timed;
▼ var k,m,n,s,a:INTT;
▼ normal(me,se): real;
▼ fun disp(me,se)=floor(normal(me,se));
▼ fun rest(i,j)=if (i>=j) then i-j else 0;
▼ fun order(i,j)=if i<=j then i else i+i-j;

```



Κεντρική σελίδα ιεραρχικού μοντέλου όπου οι παραγγελίες του πελάτη δίνονται με τη βοήθεια κανονικής κατανομής



# Συνάρτηση κόστους

$$\text{cost}(i,j,l,x) = cs * l + cbo * \text{rest}(i,l) + \text{orman}(x) * \text{order}(i,j)$$

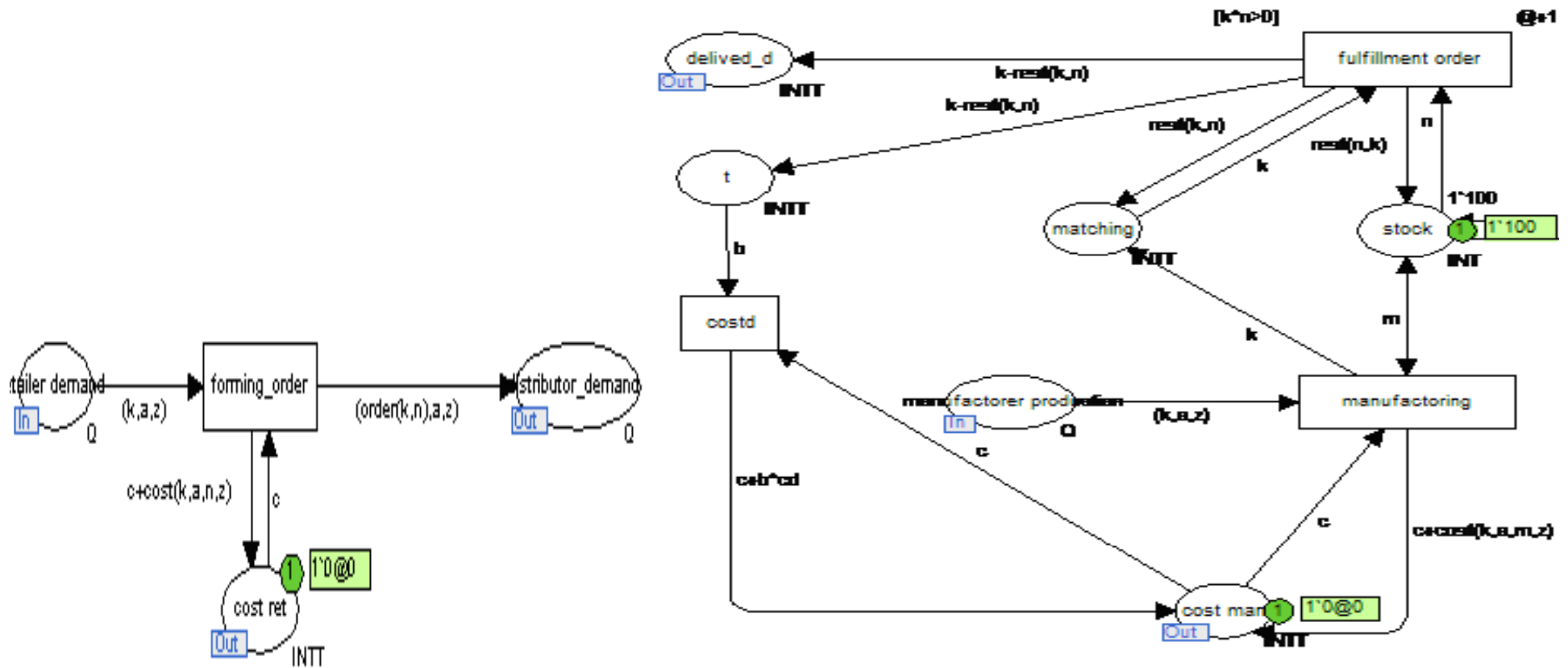
cs=12: Το κόστος αποθήκευσης ανά μονάδα προϊόντος.

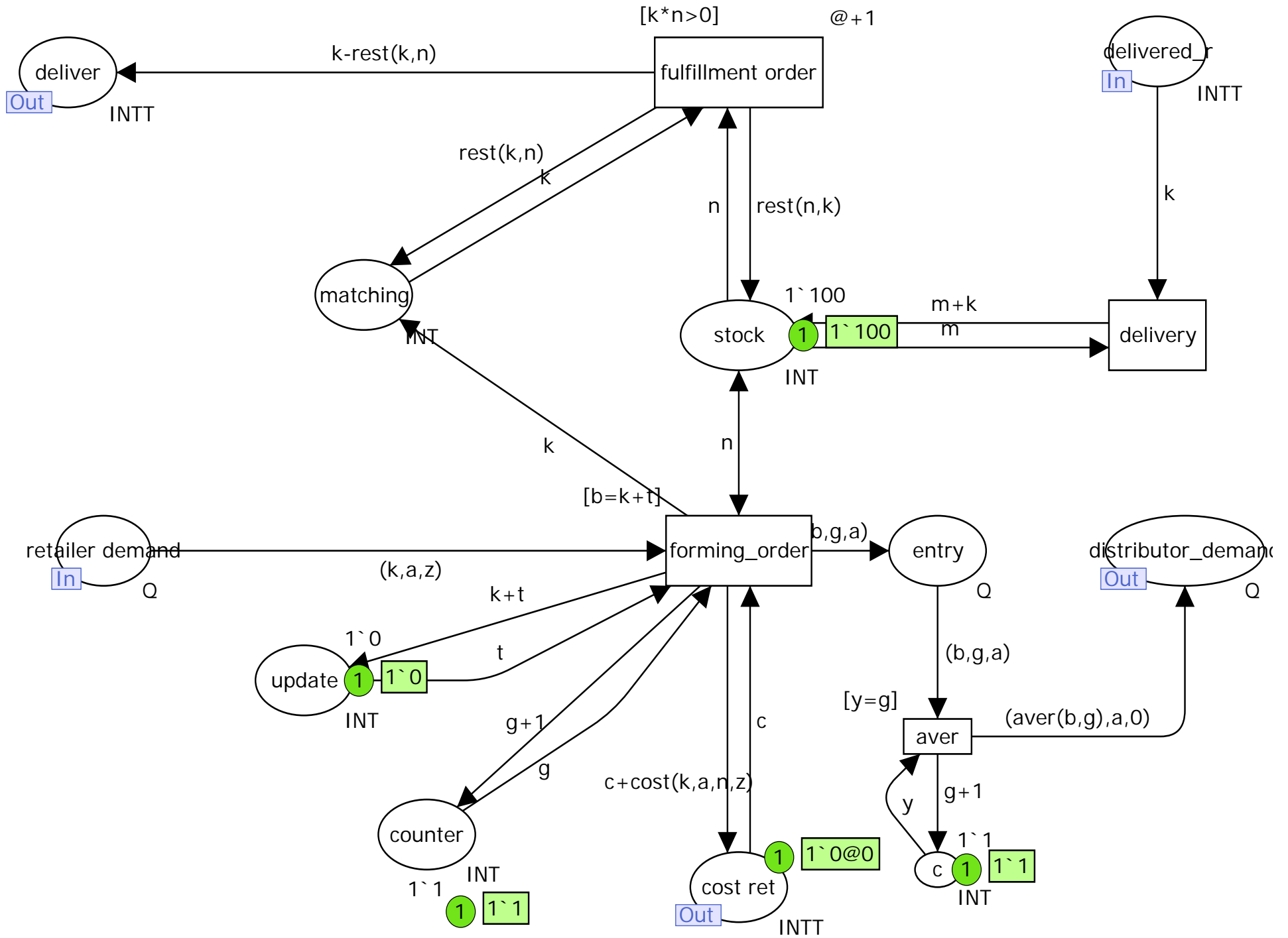
cbo=50: Το κόστος ελλείμματος της αποθήκης (backorder) ανά μονάδα προϊόντος.

co=9: Το κόστος παραγγελίας ανά μονάδα προϊόντος

cd=20: Το κόστος μεταφοράς ανά μονάδα προϊόντος.

cman=9: Το κόστος κατασκευής ανά μονάδα προϊόντος.







*Η μέθοδος κινητού μέσου χρησιμοποιείται κυρίως:*

- Σε μεγάλες περιόδους
- Σε συστήματα όπου χαρακτηρίζονται από περιοδικότητα και επανάληψη
- Σε αλυσίδες όπου οι παραγγελίες δεν έχουν μεγάλες αποκλίσεις

*Η μέθοδος εκθετικής εξομάλυνσης χρησιμοποιείται κυρίως:*

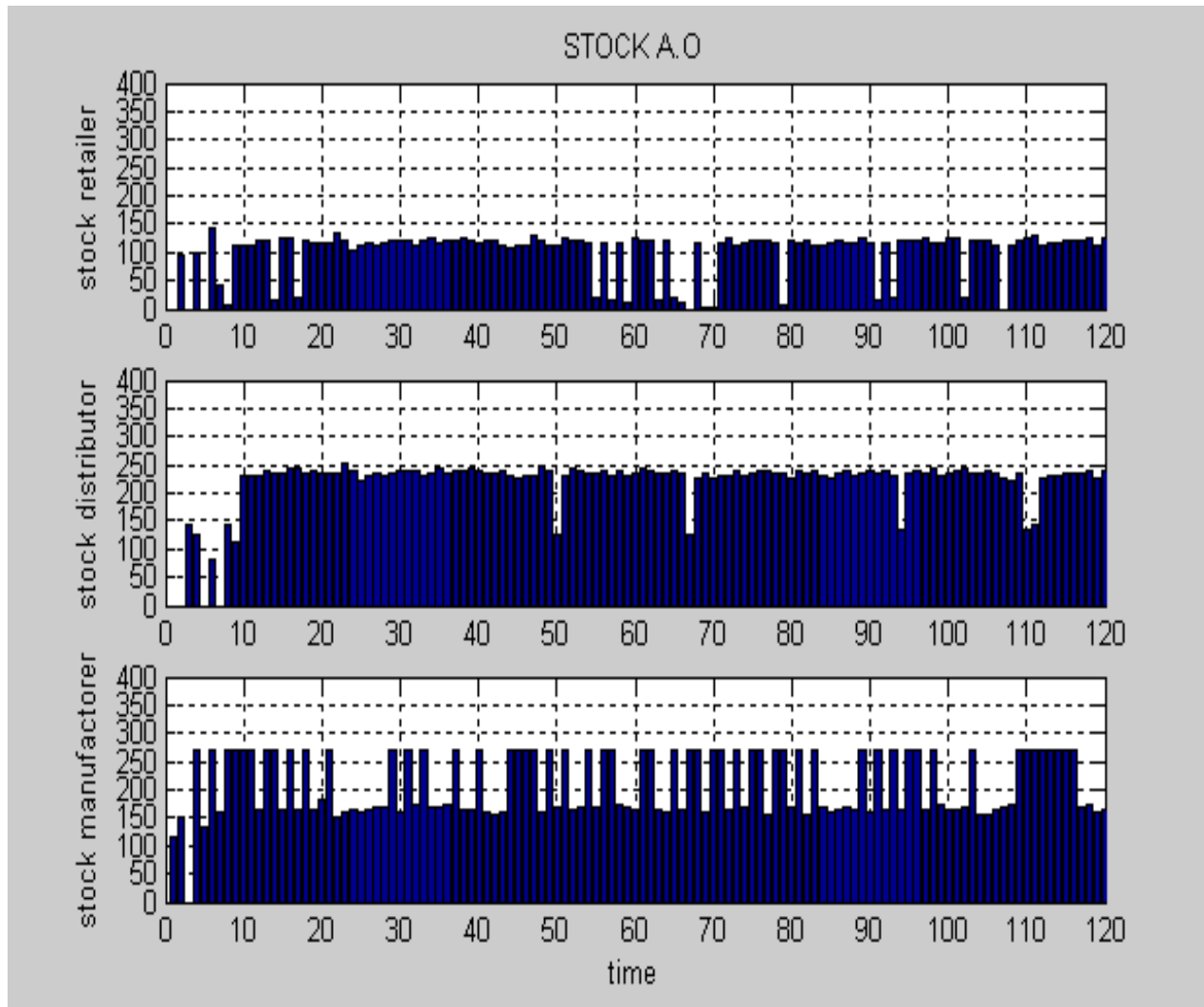
- Σε μικρό αριθμό βημάτων (π.χ μηνιαίος, βδομαδιαίος)
- Σε απότομες εναλλαγές της διακύμανσης των ποσοτήτων που χαρακτηρίζονται ως εποχιακά
- Όταν δεν είναι εφικτό το μέλος να γνωρίζει όλες τις προηγούμενες παραγγελίες αλλά κυρίως τις πρόσφατες

## Στοχαστική ανατροφοδότηση

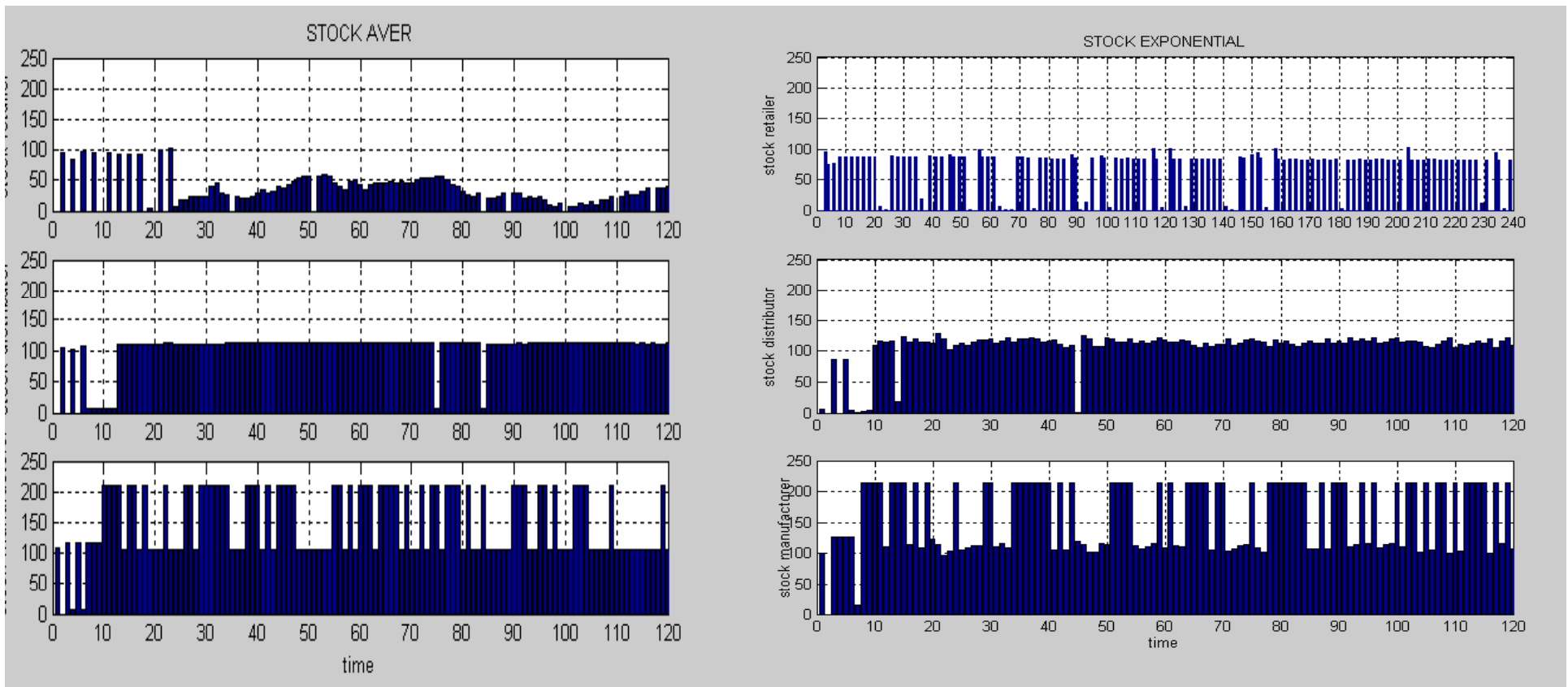
Σε αυτού του είδους την ανατροφοδότηση οι παραγγελίες δεν πραγματοποιούνται σε κάθε βήμα μιας συναλλαγής αλλά ανάλογα με το επίπεδο των αποθεμάτων ή άλλων παραγόντων

## 4.ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ

- *Κατάσταση αποθεμάτων*



Έντονες διακυμάνσεις κυρίως στα ακραία μέλη της αλυσίδας εξαιτίας του **Bullwhip**.



### Απόθεμα “aver”

- Μείωση στάθμης αποθέματος
- Μικρότερες αυξομειώσεις

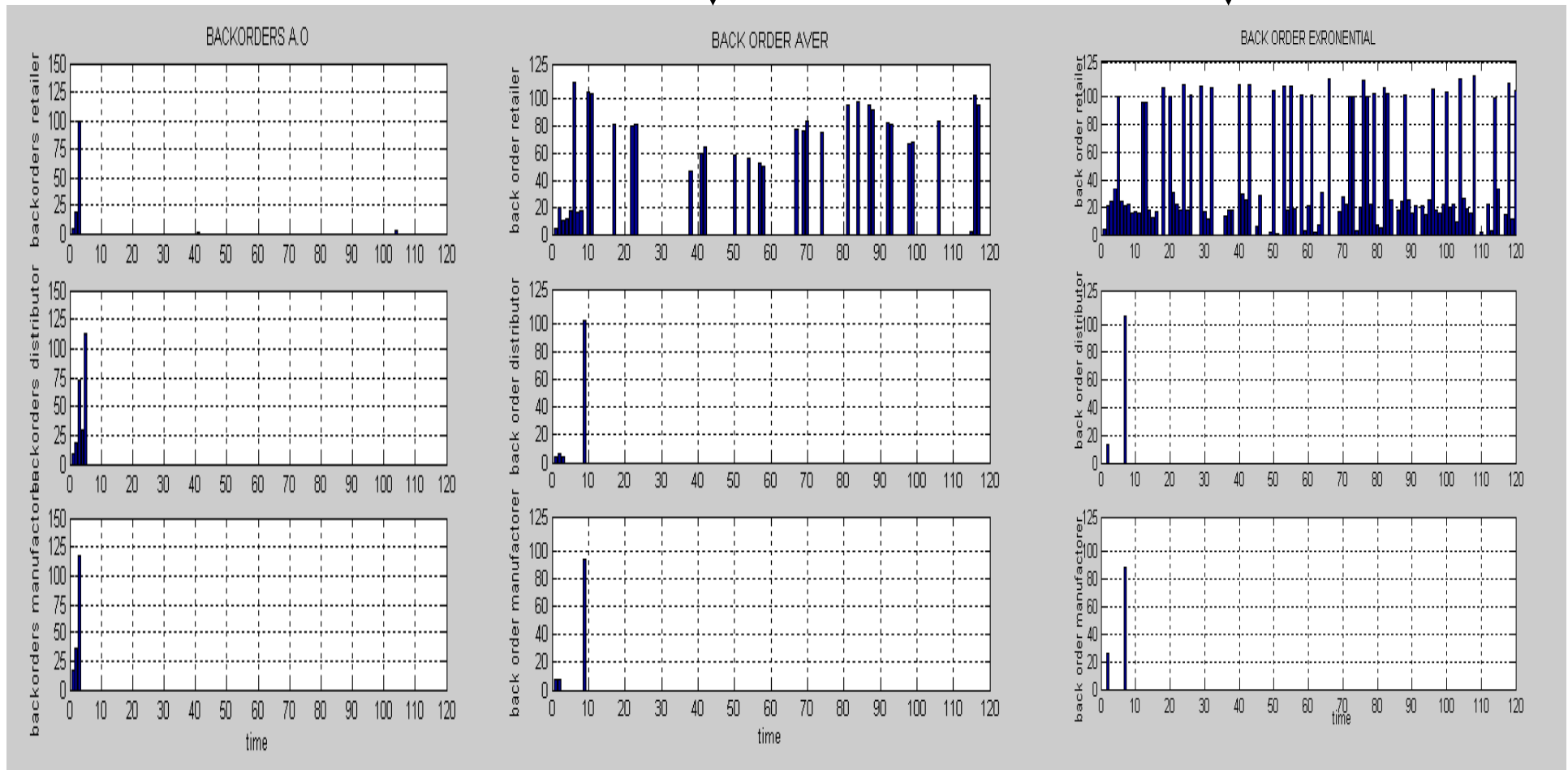
### Απόθεμα “exp”

- Μεγάλες διακυμάνσεις στη θέση του λιανέμπορου (άδειασμα-επαναφορά)

# • Κατάσταση *backorders*

- Αύξηση στον λιανέμπορο
- Μείωση στα άλλα δύο μέλη

- Μεγαλύτερη αύξηση στον λιανέμπορο
- Ίδια εικόνα για τους υπόλοιπους





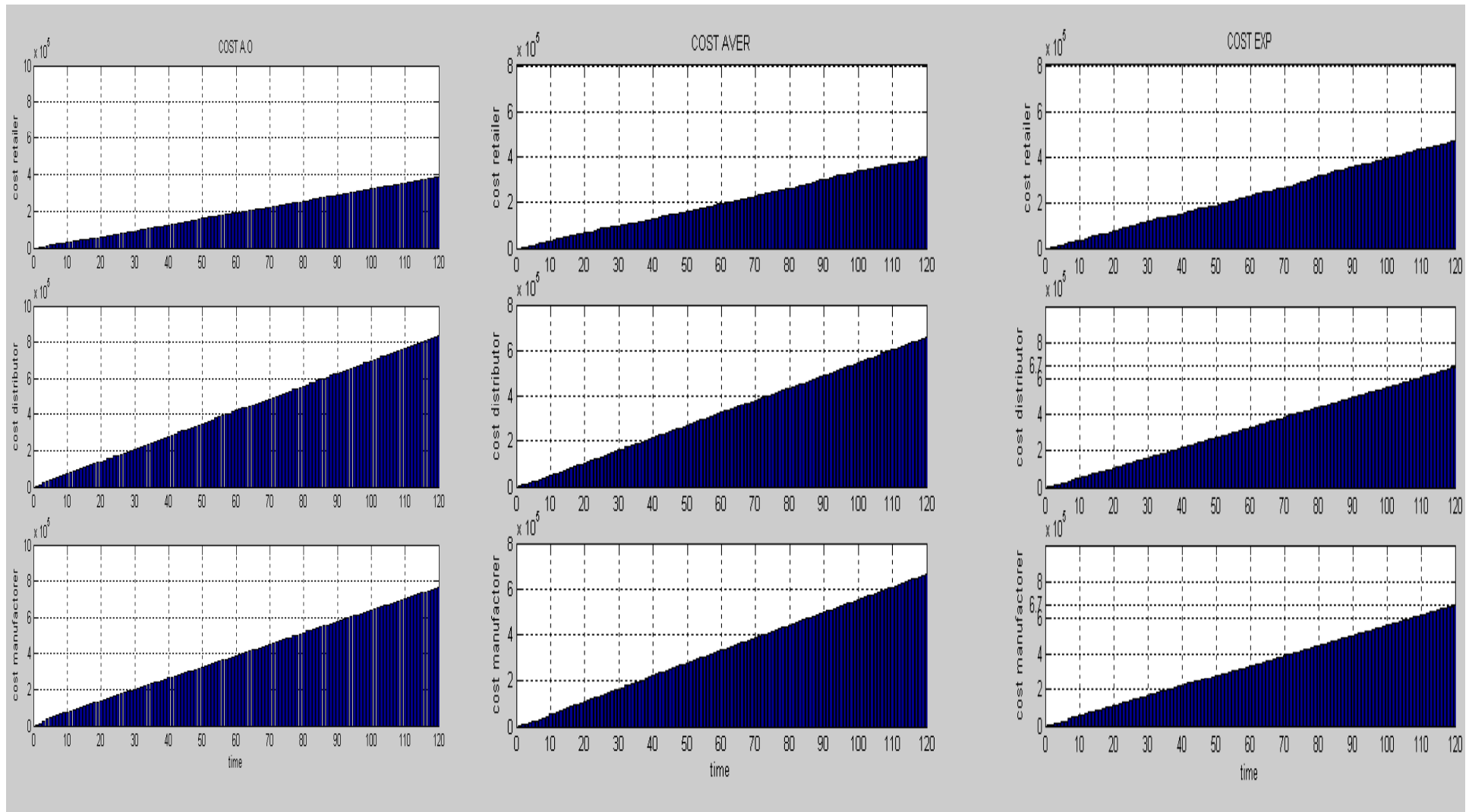
- *Κόστος*

Στον παρακάτω πίνακα καταγράφεται το συνολικό κόστος του κάθε μέλους για τις 3 διαφορετικές μεθόδους σε χρηματικές μονάδες.

	Λιανέμπορος	Μεσολαβητής	Κατασκευαστής
Aggressive ordering	387.135	834.631	765.749
Moving average	399.399	658.165	663.635
Exponential smoothing	468.955	664.004	671.408

- Για τον λιανέμπορο: Οικονομικότερη μέθοδος είναι η ΑΟ με ελάχιστη διαφορά από την ΑVER.
- Για τον μεσολαβητή: Οικονομικότερη μέθοδος είναι η ΑVER με μεγάλη διαφορά από την ΑΟ
- Για τον κατασκευαστή παρατηρείται η ίδια εικόνα.

Στα παρακάτω σχεδιαγράμματα δίνεται μία εικόνα του κόστους των μελών της αλυσίδας σε όλη την διάρκεια της διαδικασίας αθροιστικά.

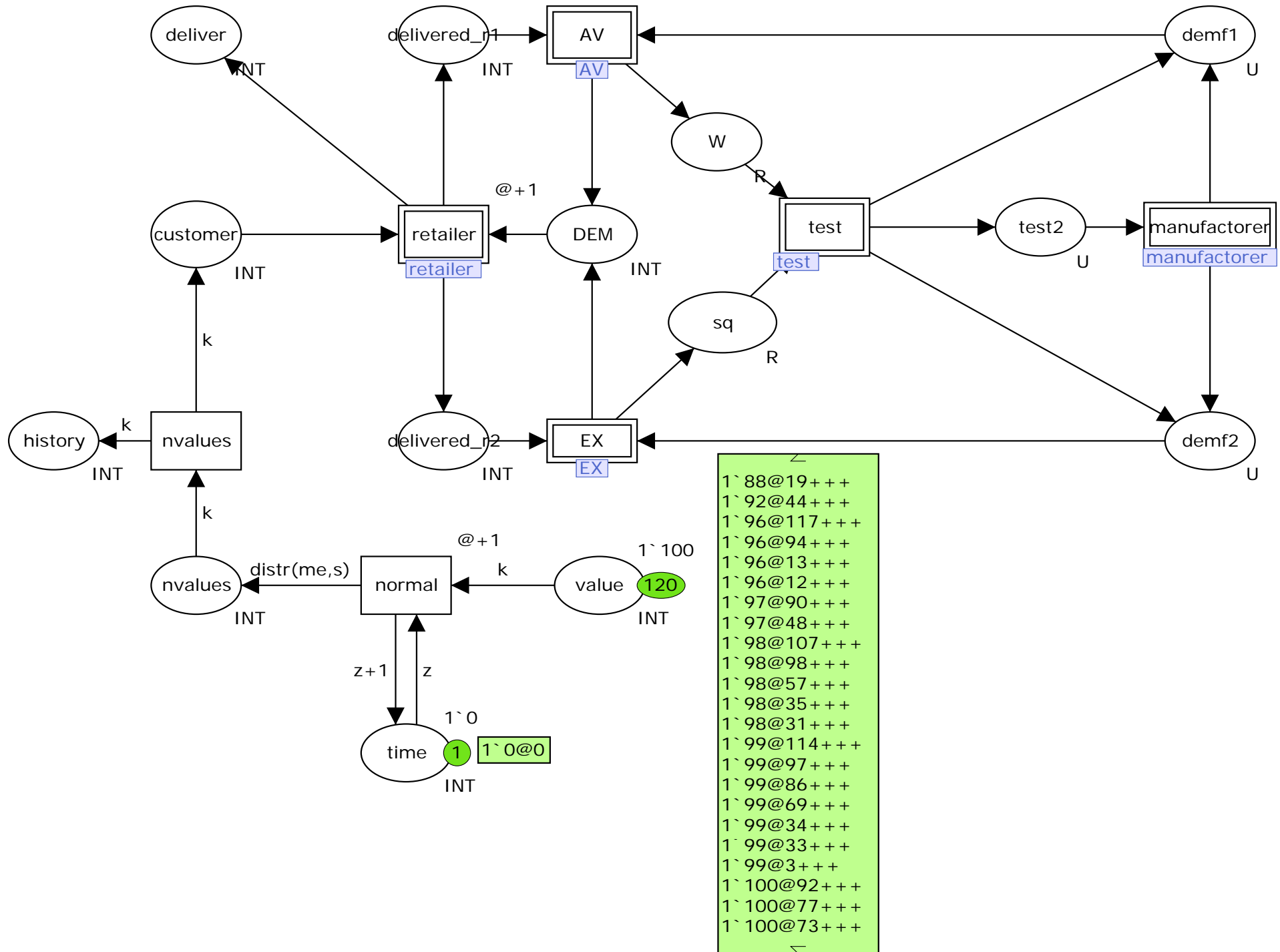


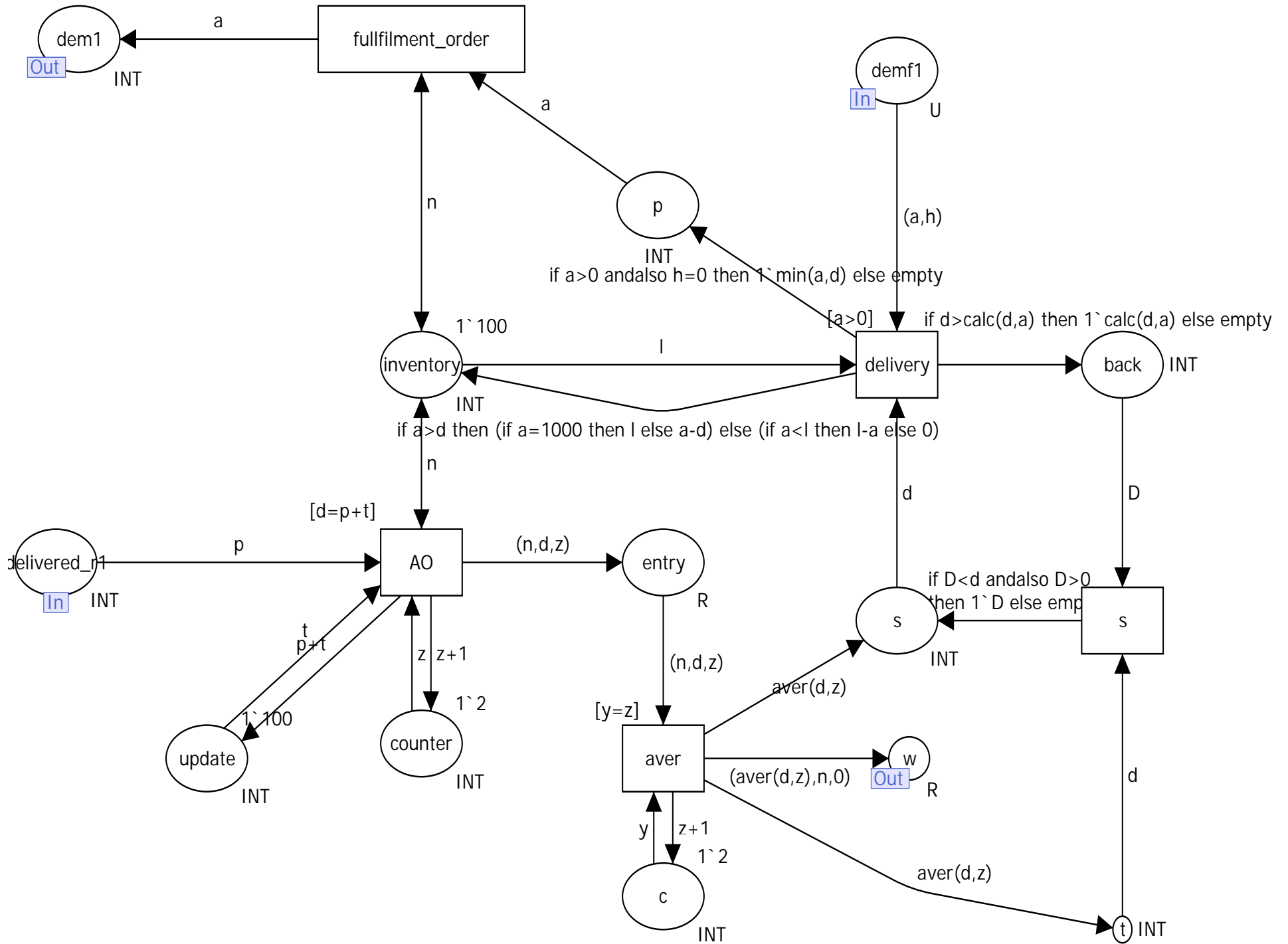
## *Συμπεράσματα*

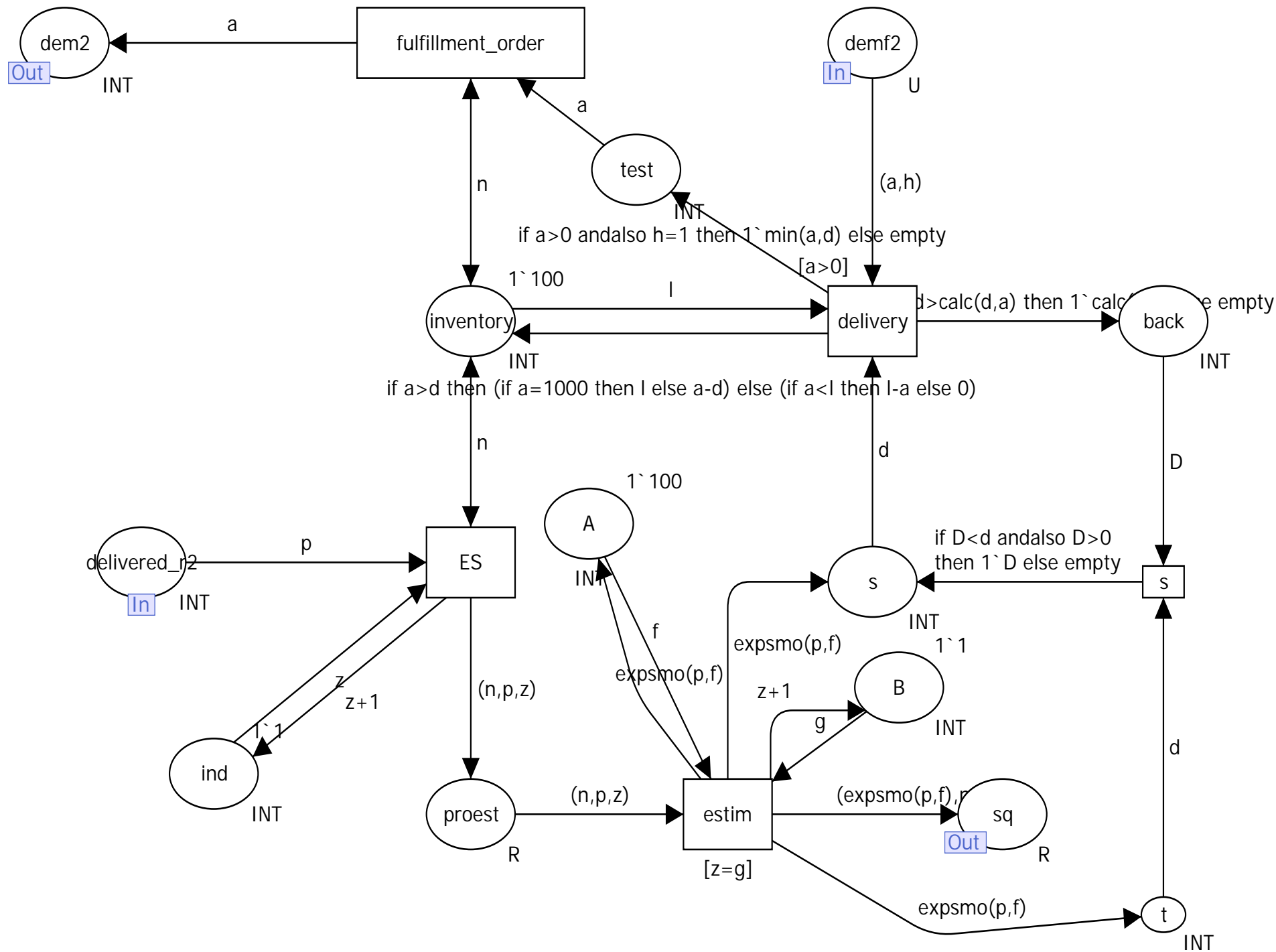
- Το συνολικό κόστος της αλυσίδας μειώθηκε με χρήση της συνάρτησης AVER (1.721.199 χ.μ) έναντι της EXP (1.804.367χ.μ) και της AO(1.987.515χ.μ)
- Με χρήση της AVER ως προς το απόθεμα και τα back orders το σύστημα είναι πιο σταθερό.
- Εξομάλυνση του φαινομένου bullwhip με χρήση της AVER.
- Τα συμπεράσματα διαφοροποιούνται λίγο αν αναλύσουμε τα αποτελέσματα για το κάθε μέλος ξεχωριστά (π.χ για τον λιανέμπορο).

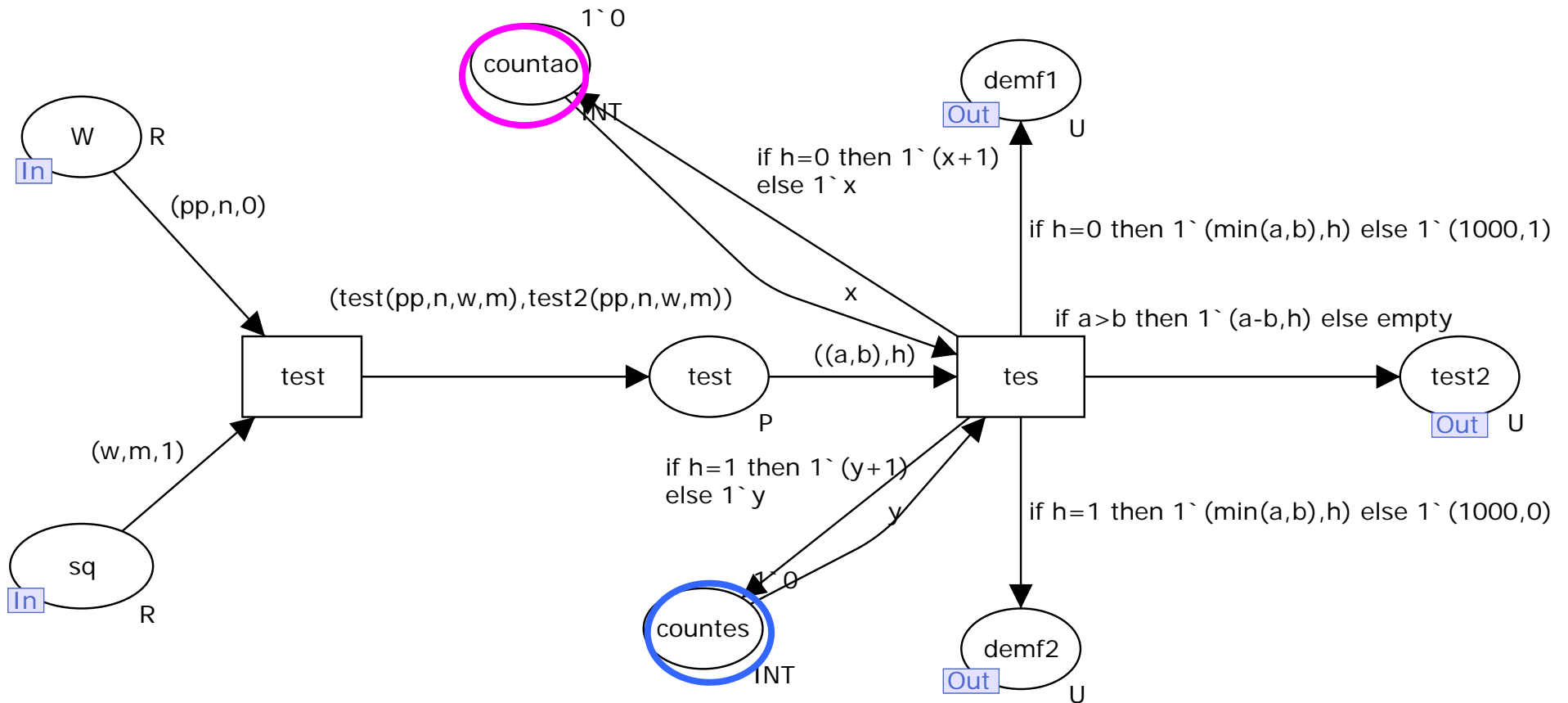
## 5.Ανταγωνιστικό μοντέλο

Σκοπός σε αυτό το κεφάλαιο είναι η δημιουργία ενός ανταγωνιστικού μοντέλου στο οποίο θα συμμετέχουν δύο μεσολαβητές (AV-EX) οι οποίοι θα παραγγέλνουν με χρήση των συναρτήσεων  $AVER$  και  $EXP$  αντίστοιχα. Με τη βοήθεια ενός test θα επιλέγεται ο ένας από τους δύο για να συνεχίσει την παραγγελία.









**Κριτήριο επιλογής:** Ο νικητής θα είναι αυτός που θα είναι σε θέση να διαθέσει άμεσα (σε συντομότερο χρόνο) τη μεγαλύτερη ποσότητα προϊόντων.

```

fun st(i,j,r,s)=if i=max(i,r) then j else s;
fun test(i,j,r,s)=if max(i,r)>st(i,j,r,s) then (if min(i,r)<st(r,j,i,s)
then (min(i,r),st(r,j,i,s)) else (if j=s then (max(i,r),j) else (st(j,i,s,r),max(j,s))))
else (max(i,r),st(i,j,r,s));
fun test2(i,j,r,s)= if test(i,j,r,s)=(i,j) then 0 else 1;

```



## ΣΤΙΣ ΠΕΡΙΠΤΩΣΕΙΣ ΟΠΟΥ:

- Το αίτημα του πελάτη είναι μικρότερο από το απόθεμα των μεσολαβητών

Επιλέγεται αυτός με το μεγαλύτερο απόθεμα

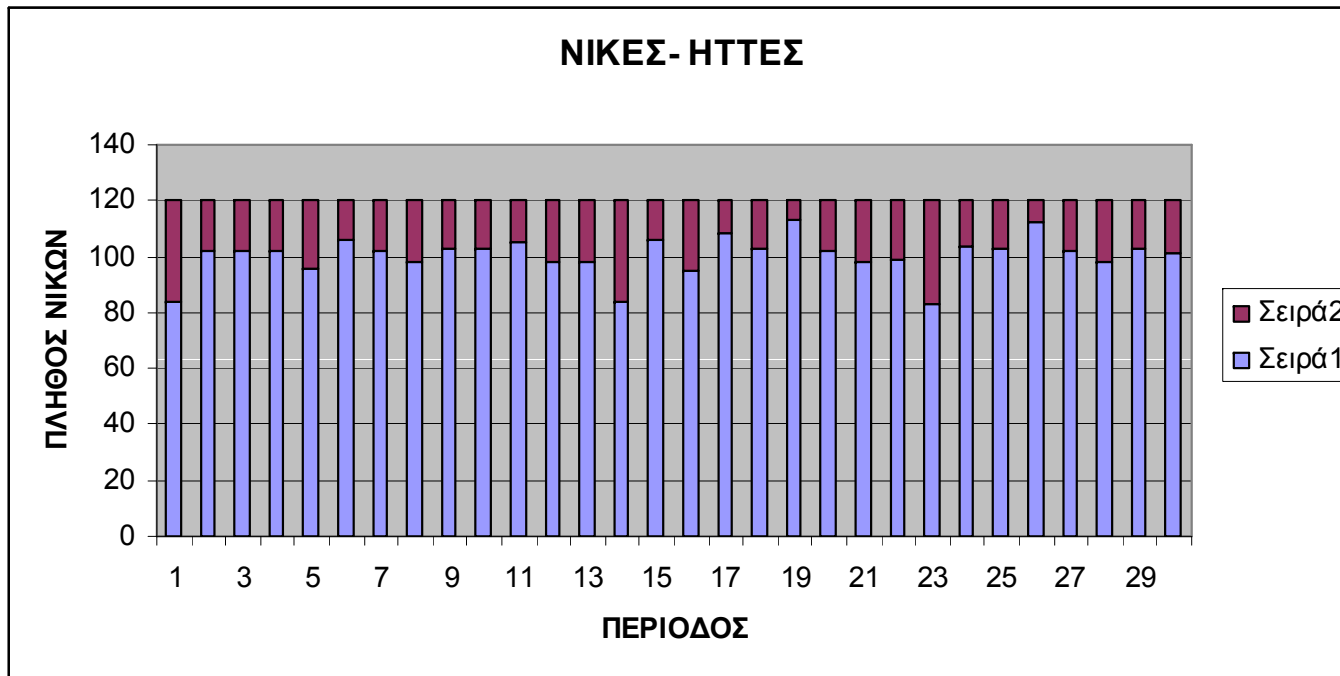
- Το αίτημα είναι μεγαλύτερο από το απόθεμα και των δύο

Επιλέγεται αυτός που θα δώσει μεγαλύτερη εξερχόμενη παραγγελία

- Το αίτημα είναι μικρότερο μόνο από το απόθεμα του μέλους (1)

Επιλέγεται το μέλος (1) αφού μπορεί να εκπληρώσει σε συντομότερο χρονικό διάστημα το αίτημα του πελάτη, ανεξαρτήτου εξερχόμενης παραγγελίας

Το πείραμα εκτελέστηκε 30 φορές. Στο παρακάτω σχήμα καταγράφονται το πλήθος των επιλογών των μεσολαβητών για τις 30 περιόδους :



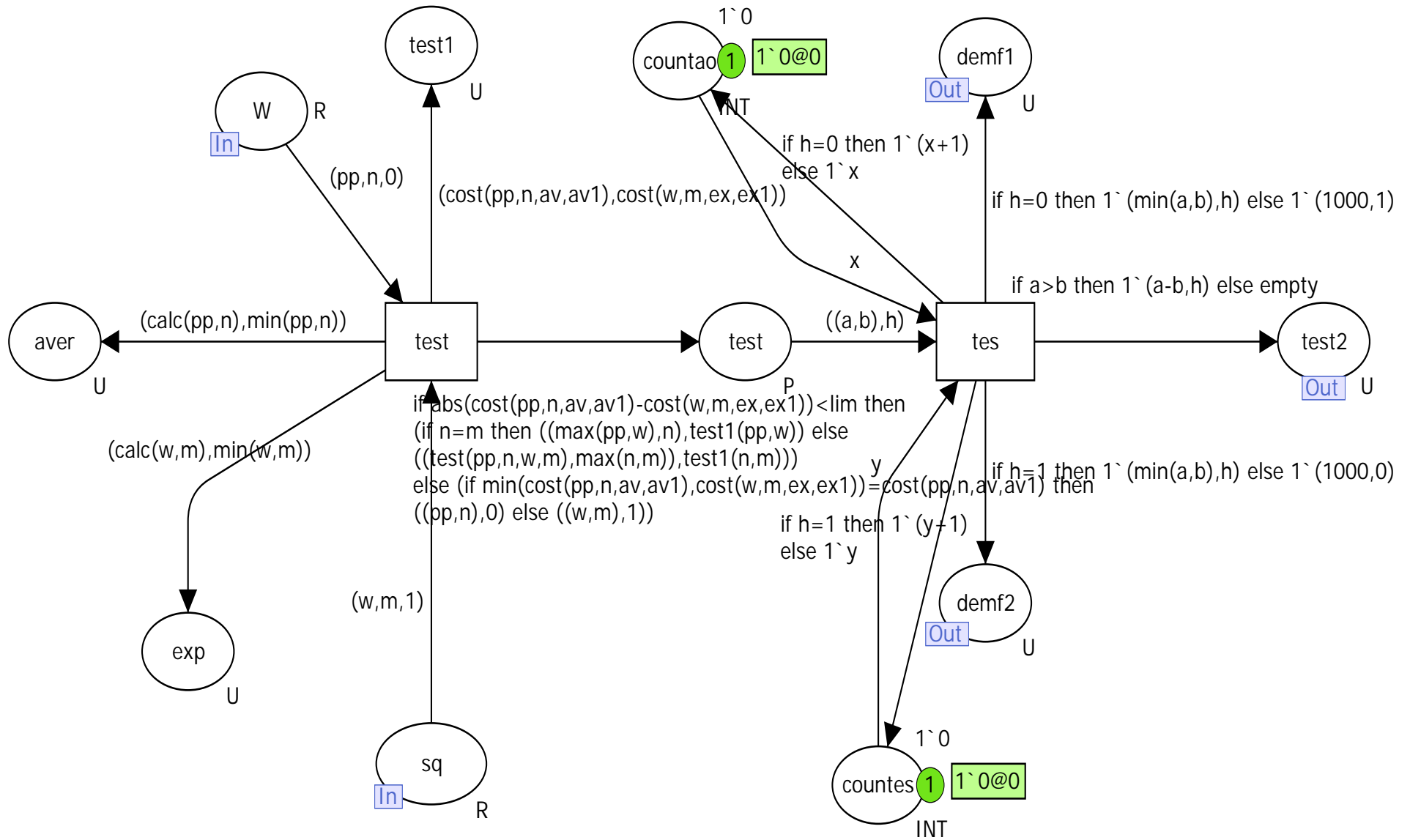
Με μπλε: επιλέγεται ο AV  
Με κόκκινο: επιλέγεται ο EX

## Ανταγωνιστικό μοντέλο με συνάρτηση κόστους

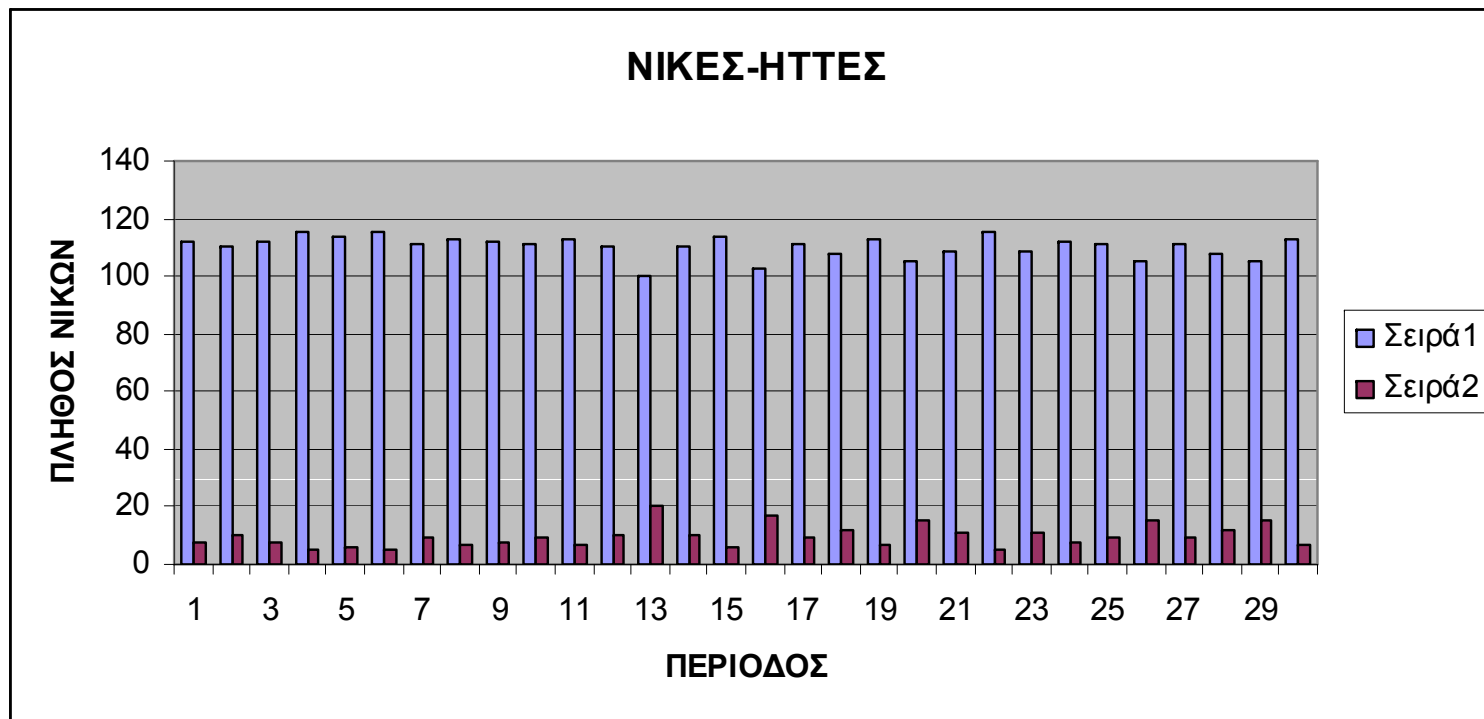
Παρακάτω γίνεται έλεγχος του αποτελέσματος με τη βοήθεια συνάρτησης κόστους  
$$\text{fun cost}(i,j,k,l)=l*\text{calc}(i,j)+k*\text{min}(i,j);$$

όπου “calc” η ποσότητα που θα παραγγείλει ο μεσολαβητής που θα επιλεγεί από το τεστ από τον κατασκευαστή (όταν  $p>n$ ) και “min” η ποσότητα που θα στείλει τελικά ο νικητής.

Το κριτήριο επιλογής παραμένει το ίδιο. Τις περισσότερες φορές το μέλος που επιλέγεται θα έχει είτε μεγαλύτερο απόθεμα, είτε μεγαλύτερη εξερχόμενη τιμή και άρα σημειώνει μεγαλύτερο κόστος. Στο σημείο αυτό εισάγεται και ένας περιορισμός: δεν θα πρέπει η διαφορά στα κόστη να ξεπεράσει το επίπεδο που ορίζει η τιμή της  $lim$  (εδώ 800χ.μ.). Σε αυτή τη περίπτωση επιλέγεται το μέλος με το μικρότερο κόστος.



Μετά από 30 δοκιμές του πειράματος, καταλήγουμε στο συμπέρασμα ότι περίπου στο 90% των περιπτώσεων επιλέγεται ο AV.



Για τον λόγο αυτό προσπαθούμε να βρούμε την κατάλληλη πολιτική που θα ακολουθήσει ο ex για να **ανατρέψει το αποτέλεσμα**

Αν θέσουμε  $c_1$  το κόστος του ex και  $c_2$  το κόστος του an τότε

$$c_1 = l x_{11} + k x_{12} \text{ και } c_2 = l' x_{21} + k' x_{22} \text{ (} l' = 30 \text{ και } k' = 12 \text{)}$$

$$\text{(fun cost}(i,j,k,l) = l * \text{calc}(i,j) + k * \text{min}(i,j);)$$

όπου  $x_{11}$ ,  $x_{21}$  ποσότητες που αντιστοιχούν στην παραγγελία που θα δώσουν στον κατασκευαστή και  $x_{12}$ ,  $x_{22}$  ποσότητες που θα παραδώσουν στον λιανέμπορο.

Θα προσπαθήσουμε να μειώσουμε τις μεταβλητές  $k, l$  μέχρι να ανατραπεί το αποτέλεσμα.

Θα βρούμε την μεγαλύτερη τιμή της συνάρτησης  $c_1$  υπό τις προϋποθέσεις

$$30x_1 + 12x_2 - lx_1 - kx_2 > 800 \quad (c_2 - c_1 > 800)$$

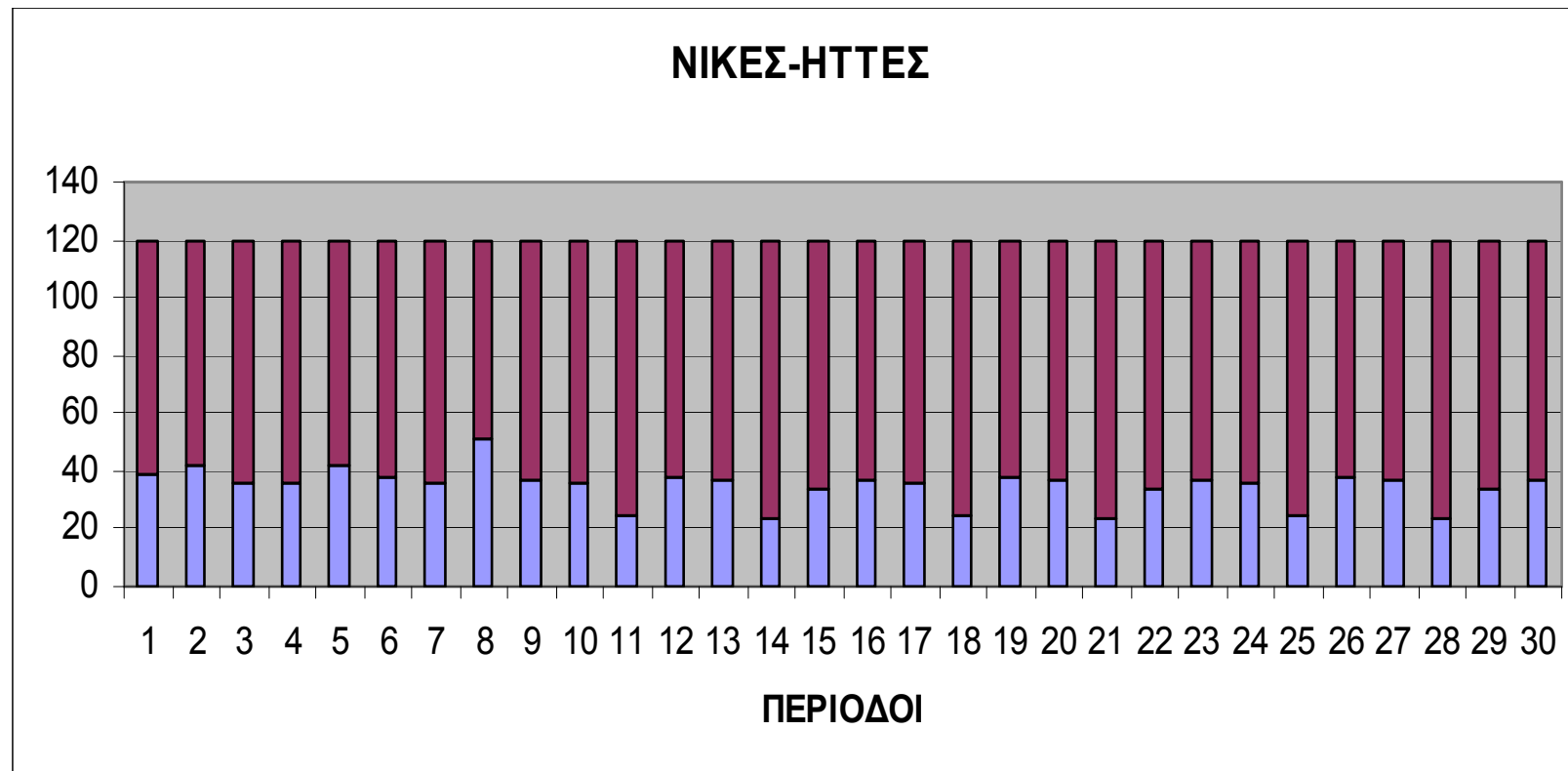
$$lx_1 - kx_2 > 0 \quad (c_1 > 0)$$

$$l \leq 30$$

$$k \leq 12$$

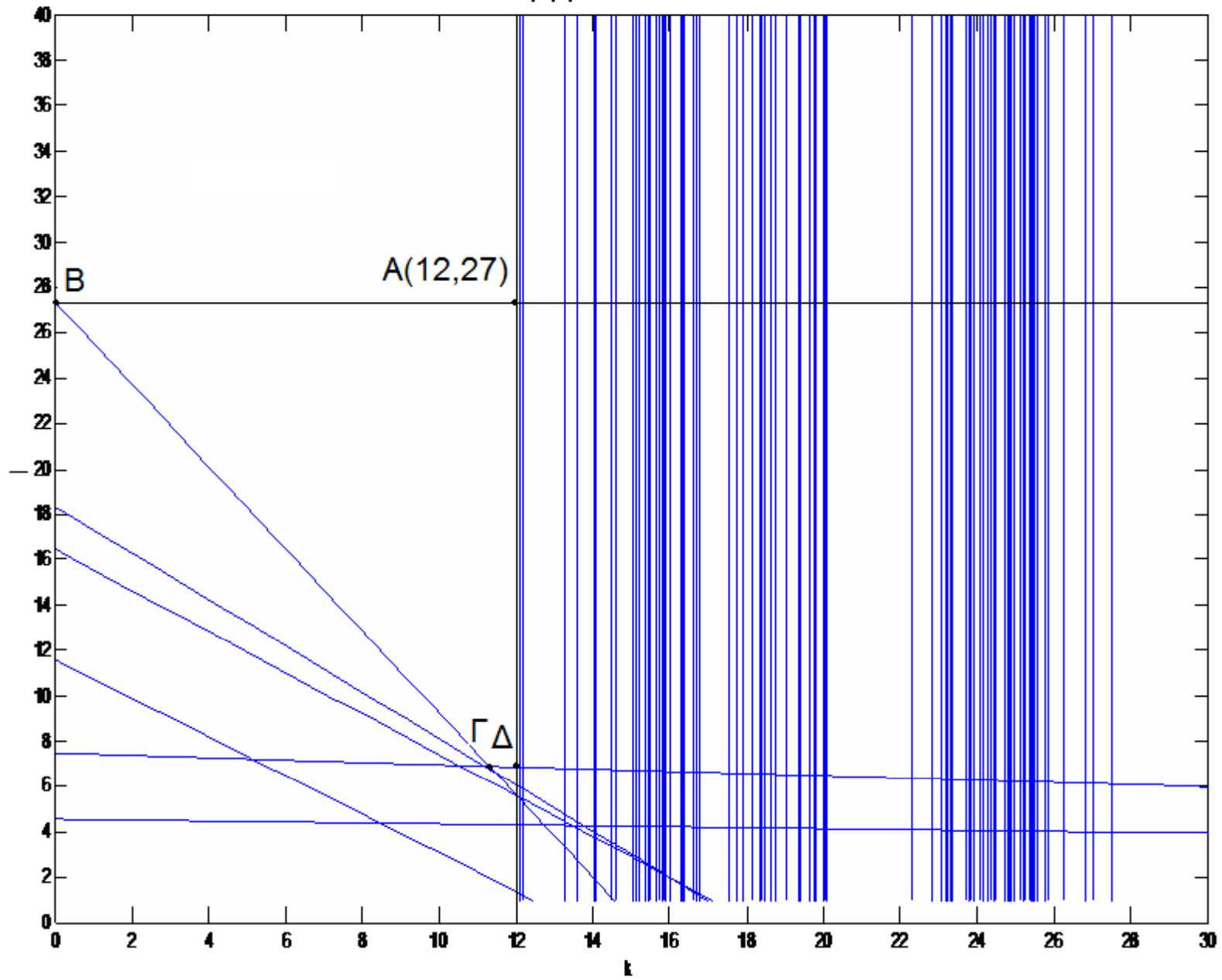
Μετά από δοκιμές στο μοντέλο για τιμές  $k=12$  και  $l < 30$  βρέθηκε ότι η μεγαλύτερη τιμή του  $l$  για την οποία ικανοποιούνται οι συνθήκες είναι για  $l=27$ , κάτι που επιβεβαιώνεται και στο σχήμα.

Τα αποτελέσματα της προσομοίωσης για τις νέες τιμές των μεταβλητών  $k, l$  στη συνάρτηση κόστους του  $ex$  είναι τα εξής:





Η συνάρτηση  $30x_2 + 12x_2 - kx_1 - x_2 = 800$



## **Συμπεράσματα**

- Στο ανταγωνιστικό μοντέλο όπως και στην περίπτωση που μελετήθηκαν οι συναρτήσεις  $aver$  και  $exp$  ξεχωριστά, η χρήση της πρώτης είναι αποτελεσματικότερη.
- Μία μικρή αλλαγή στις μεταβλητές των τιμών  $(k, l)$  μπορεί να αλλάξει το αποτέλεσμα στο παιχνίδι του ανταγωνισμού.
- Το αποτέλεσμα του νικητή εξαρτάται από διάφορους παράγοντες:
  - i) Τον τρόπο με τον οποίο παραγγέλνει ο πελάτης.
  - ii) Το επίπεδο όπου δεν θα πρέπει να ξεπεράσει η διαφορά των κοστών ( $lim$ )
  - iii) Το επίπεδο των αποθεμάτων

***Σας ευχαριστώ πολύ!!!***